

**Faculdade de Engenharia da Universidade do Porto**



**NeuronDyn: Live Neurotransmitter Vesicle Movement  
Dynamics in Living Neurons**

Ivo Francisco Matos da Silva

Master Thesis elaborated on the  
Integrated Master in Bioengineering  
Branch of Biomedical Engineering

Supervisor: PhD João Paulo Cunha, DEEC-FEUP/ INESC TEC  
Co-Supervisor: PhD Paula Sampaio, IBMC

September 2014



*Aos meus Pais e ao meu Irmão*



## **Abstract**

Advances in the field of Biomedical Engineering and Neuroscience allowed to study the neuronal interactions in a more precise and simple way. In this work, the transport mechanism of neurotransmitters that circulate through the neuron within vesicles is evaluated. An irregularly transport of these vesicles may indicate some neurodegenerative diseases such as Parkinson or Alzheimer, showing the importance to create effective tools to characterize it. Through the development of a semi-automatic tracking algorithm capable of detect and follow the vesicles over the video, it's possible to remove the workload from researchers that make this same tracking manually, increasing the accuracy of the analysis.

This project follows a first version of a vesicle tracking and characterization algorithm in videos obtained with confocal microscopy and performed in MATLAB. In collaboration with IBMC, where the need of the project emerged, a first phase of training was drawn in which the researcher marks a number of vesicles is sure, the same number of cases of doubt, and the position of the cell body/reference point for movement. After the training stage, and using an artificial neural network (ANN) classifier, the algorithm learns to perform the marking of vesicles to the rest of the frames. The approach used to track the vesicles is a Global Nearest Neighbour with a Hungarian algorithm to perform the correspondence between frames, which allows to link vesicles even in the presence of gaps. Some important characteristics of the vesicles are obtained and the results are saved in an Excel file.



## Resumo

Avanços na área da Engenharia Biomédica e da Neurociência permitiram estudar de uma forma mais simples e precisa as interações neuronais. Neste trabalho é abordado o mecanismo de transporte de neurotransmissores, que circulam pelos neurónios agregados em vesículas. Uma forma irregular de movimentação destas vesículas pode indiciar certas doenças neurodegenerativas como *Parkinson* ou *Alzheimer*, pelo que a produção de ferramentas eficazes para a caracterização do movimento de vesículas seja importante. Através do desenvolvimento de um algoritmo de tracking semi-automático capaz de detectar e seguir as vesículas ao longo de uma sequência de imagens, é possível retirar carga de trabalho a investigadores que fazem esse mesmo *tracking* manualmente e assim aumentar a sua precisão da análise.

Este projeto surge na sequência de uma primeira versão de um algoritmo de tracking e caracterização das vesículas marcadas em vídeos de microscopia confocal, efetuado em MATLAB. Em colaboração com o IBMC, de onde surgiu a necessidade do projeto, foi desenhada uma primeira fase de treino em que o investigador marca um determinado número de vesículas que tem a certeza e ainda o mesmo número de casos de dúvida, bem como a posição do corpo celular/ponto de referência do movimento. Após o treino, e através de um classificador de redes neuronais (ANN), o algoritmo aprende a efetuar a marcação das vesículas para o resto das frames. A abordagem de tracking utilizada é um Global Nearest Neighbor com um Hungarian Algorithm para efetuar a correspondência entre frames, que permite associar as vesículas mesmo na presença de *gaps*. São então obtidas algumas características das vesículas consideradas importantes, sendo o resultado gravado num ficheiro Excel.



# Contents

<b>List of Figures .....</b>	<b>xiii</b>
<b>List of Tables .....</b>	<b>xvii</b>
<b>List of Acronyms .....</b>	<b>xix</b>
<b>Chapter 1 .....</b>	<b>1</b>
Introduction.....	1
1.1 - Scope of the Thesis.....	2
1.2 - Motivations.....	3
1.3 - Main Objectives .....	4
1.4 - Publications .....	5
1.5 - Work Structure .....	5
 <b>Part I - State of the Art.....</b>	 <b>7</b>
 <b>Chapter 2 .....</b>	 <b>9</b>
Neuronal Exchanges .....	9
2.1 - Neuroscience .....	9
2.2 - Neurons .....	10
2.2.1 - Anatomy .....	10
2.2.2 - Electrical Potential.....	11
2.2.3 - Synapses .....	12
2.2.4 - Neurotransmitters .....	13
2.3 - Neuronal Intracellular Transport.....	13
 <b>Chapter 3 .....</b>	 <b>19</b>
Neuroimaging Techniques.....	19
3.1 - Neuroimaging.....	19
3.2 - Optical vs Electron Microscopy .....	20
3.3 - Confocal Microscopy .....	21
3.3.1 – Laser Scanning Confocal .....	22

3.3.2 – Spinning Disk Confocal .....	23
<b>Chapter 4 .....</b>	<b>25</b>
Neuroimage Processing and Analysis.....	25
4.1 - CAD System.....	25
4.2 - Image Basics.....	26
4.3 - Image Restoration/Denoising .....	27
4.3.1 – Noise Models .....	28
4.3.2 – Spatial Domain.....	30
4.3.3 – Transform Domain.....	31
4.4 - Image Enhancement .....	32
4.4.1 –Spatial Domain.....	32
4.4.2 – Frequency Domain.....	33
4.5 - Image Segmentation .....	33
4.5.1 – Thresholding Methods .....	34
4.5.2 – Image Based.....	35
4.5.3 – Model Based .....	37
4.6 - Feature Extraction .....	41
4.7 - Clustering and Classification.....	42
4.7.1 – Unsupervised Methods.....	43
4.7.2 –Supervised Methods.....	45
4.8 - Tracking/Motion Analysis in CAD Systems.....	48
4.8.1 – Tracking Methods .....	48
4.8.2 – Vesicle Tracking .....	52
4.8.3 – Manual vs Automatic Tracking .....	53
4.8.4 – Particle Tracking Tools.....	53
4.8.5 – Tracking Measures.....	53
4.9 - Evaluation.....	56
 <b>Part II - <i>NeuronDyn</i> - Vesicle Movement Characterization Software .....</b>	 <b>59</b>
 <b>Chapter 5 .....</b>	 <b>61</b>
<i>NeuronDynamics</i> .....	61
5.1 - <i>NeuronDynamics</i> : A Method for Neurotransmitter Vesicle Movement Characterization in Neurons .....	61
5.1.1 – Image Acquisition.....	61
5.1.2 – Algorithm Training .....	62
5.1.3 – Vesicle Segmentation and Classification.....	62
5.1.4 – Vesicle Tracking .....	62
5.1.5 – Algorithm Evaluation.....	63
5.2 - Discussion and Improvements.....	64
 <b>Chapter 6 .....</b>	 <b>67</b>
<i>NeuronDyn</i> .....	67
6.1 - Introduction .....	67
6.2 - Problem, Challenges and Difficulties.....	68

6.3 - NeuronDyn Modules .....	68
6.3.1 – Dataset Videos .....	70
6.3.2 – Input Videos to NeuronDyn.....	71
6.3.3 – Parameters Selection.....	72
6.3.4 – Algorithm Training.....	75
6.3.5 – Vesicle Segmentation and Classification.....	76
6.3.6 – Vesicle Tracking.....	76
6.3.7 – Algorithm Evaluation .....	77
6.4 - Results and Discussion.....	78
<b>Chapter 7.....</b>	<b>87</b>
Conclusions.....	87
References.....	89
<b>Appendix .....</b>	<b>95</b>
Appendix A.....	97
Cell Tracking Challenge .....	97
Appendix B.....	107
CellREM: The Recursive Error Metric For Cell Tracking.....	107
Appendix C.....	115
PGBiomed 2014 Poster Presentation.....	115



## List of Figures

Figure 1.1 – Main steps of the vesicle’s Neuroscience research. ....	2
Figure 2.1 – Representation of neuron’s anatomy. [19]. ....	11
Figure 2.2 – Material moving in fast axonal transport, in both anterograde and retrograde directions. [39].....	15
Figure 2.3 – Movement of cytoskeletal material in slow axonal transport, in anterograde direction. [39].....	15
Figure 2.4 – Neuronal types of vesicles. [47] .....	16
Figure 3.1 – Comparison of the spatial and temporal resolution and scope of some imaging techniques used in Neuroscience. [56] .....	20
Figure 3.2 – Main techniques in Microscopy field.....	21
Figure 3.3 – Comparison between a conventional, fluorescence and confocal microscope. [59] .....	22
Figure 3.4 – Differences between a) CLSM and b) Spinning Disk Confocal Microscopy techniques. [61].....	23
Figure 4.1 – Typical workflow of a Tracking CAD System. ....	26
Figure 4.2 – Image Engineering main levels. [64] .....	26
Figure 4.3 – Image Denoising techniques Classification. Adapted from [66] .....	30
Figure 4.4 – Image Enhancement methods.....	32
Figure 4.5 – Image Segmentation overview. ....	34

Figure 4.6 – Histogram based thresholding methods. [65].....	35
Figure 4.7 – Region Growing Segmentation method. [76].....	36
Figure 4.8 – Split and Merge Segmentation method. [77].....	37
Figure 4.9 – Watershed method. [78] .....	37
Figure 4.10 – Example of snake application, showing the stages between iterations. [79].....	38
Figure 4.11 – Graph cut segmentation. [80] .....	39
Figure 4.12 – Voxel Atlas based Segmentation method. [71] .....	40
Figure 4.13 – Overview of shape descriptors. [83].....	42
Figure 4.14 – Machine Learning methods .....	42
Figure 4.15 – After region detection, specific features are extracted and grouped. ....	43
Figure 4.16 – K-means clustering example. ....	44
Figure 4.17 – Historical popularity of some machine learning classification algorithms. [88].....	45
Figure 4.18 – Interaction between layers in a Neural Network .....	46
Figure 4.19 – SVM classifier [91] .....	47
Figure 4.20 – Multiple moving objects tracking illustration [92].....	48
Figure 4.21 – Object tracking approaches. [93].....	49
Figure 4.22 – Hungarian algorithm association. [95] .....	51
Figure 4.23 –Illustration of object association over time with some dynamic problems. [99] .....	52
Figure 4.24 – Detected objects with four types of labels. [104].....	56
Figure 4.25 – ROC Curve representation [105].....	57
Figure 5.1 – <i>NeuronDynamics</i> pipeline. ....	64
Figure 5.2 – Algorithm Performance Comparison for <i>NeuronDynamics</i> and <i>FluoTracker</i> using default settings. ....	65
Figure 5.3 – <i>NeuronDynamics</i> Interface in Matlab. ....	66
Figure 5.4 – Example of <i>NeuronDynamics</i> results. ....	66
Figure 6.1 – <i>NeuronDyn</i> pipeline.....	69

Figure 6.2 – <i>NeuronDyn</i> UI.....	70
Figure 6.3 – Upload videos interface in <i>NeuronDyn</i> .....	71
Figure 6.4 – Characteristics and display of the selected videos .....	72
Figure 6.5 – Tracking parameters of <i>NeuronDyn</i> .....	73
Figure 6.6 – Path Detection pipeline. ....	74
Figure 6.7 – Process selection and vesicle training detection.....	74
Figure 6.8 – <i>NeuronDyn</i> Training section. ....	75
Figure 6.9 – Neural Network structure used for classification.....	76
Figure 6.10 – Original and Path images obtained for datasets 1,2 and 3.....	78
Figure 6.11 – Path detection for the first dataset using only 5 frames .....	79
Figure 6.12 – Intensity profile of an image from Dataset 3.....	79
Figure 6.13 – Detected vesicles in the first frame by <i>NeuronDyn</i> (right image), based in the training choices (left image). The training frame selected was the first one. ....	80
Figure 6.14 – Histogram with an example of the user’s choices of 10 true vesicles (at right) and 10 ambiguous vesicles (at left), for Dataset 3.....	80
Figure 6.15 – Comparison between <i>NeuronDyn</i> segmentation, without interference of the background information, and a case with interference of the background...	81
Figure 6.16 – Classification performed by <i>NeuronDyn</i> between marked points, where features 1 is the maximum intensity value, and feature 2 the major axis length. Real vesicles marked by the user are shown as red points and non- vesicles as blue points.....	81
Figure 6.17 – Detected vesicles and the tracks created to Dataset 3. The expansion of a portion of the image allows to verify the detected vesicles in that zone, and the tracks associated.....	82
Figure 6.18 – Tracking results section of <i>NeuronDyn</i> UI, with the tracks obtained for Dataset 3 .....	83
Figure 6.19 – Image from “View all Tracks” (left image) and a specific case of “View Vesicle” (right image), with the detected positions linked and a reference to the corresponding frame number.....	83
Figure 6.20 – Example of Excel results for Dataset 3, considering only 10 frames.....	84
Figure A.1 – Datasets used in the Cell Tracking Challenge.....	100

Figure A.2 – Overview of the Cell Tracking Algorithm. ....	100
Figure A.3 – Linear expansion of the histogram. ....	101
Figure A.4 – Example of a) initial image, b) after denoising and c) after enhancement. ....	101
Figure A.5 – Illustration of Canny Edge Segmentation.....	102
Figure A.6 – Cells after a) Enhancement method, b) Canny Segmentation, c) Edge linking and morphological operators. ....	102
Figure A.7 – Example of image obtained and the .txt of the dataset. ....	103
Figure A.8 – Example of a division detected by the algorithm . ....	104
Figure A.9 – Example of a merge marked by the algorithm. ....	104
Figure A.10 – Results from segmentation and labelling.....	105
Figure B.1 – At left Dataset 1 with 160x82 pixels, and at right Dataset 2 with 640x400 pixels .....	111

## List of Tables

Table 2.1 — Axonal Transport : types, velocity and carried substances. Adapted from [46] .....	14
Table 2.2 — Diseases related to the NIT. [55] .....	17
Table 4.1 — Main types of noise in images, its distribution formula, representation and an example .....	29
Table 4.2 — Comparison between different moving object tracking techniques .....	50
Table 4.3 — List of some available particle tracking tools. Adapted from [79] .....	55
Table 6.1 — Datasets used for testing <i>NeuronDyn</i> .....	71
Table 6.2 — Evaluation of <i>NeuronDyn</i> Segmentation process. ....	85
Table B.1 — Tracking accuracy results using different frame rates .....	112



## List of Acronyms

2DG	2-Deoxy-D-glucose
ALS	Amyotrophic lateral sclerosis
ANN	Artificial Neural Network
AP	Action Potential
ATP	Adenosine Triphosphate
CAD	Computer Aided Diagnosis
CCV	Color Coherence Vectors
CLSM	Confocal Laser Scanning Microscopy
CM	Color Moments
CNS	Central Nervous System
DCV	Dense Core Vesicles
DEEC	<i>Departamento de Engenharia Electrotécnica e de Computadores</i>
DIC	Differential Interference Contrast
DoG	Difference of Gaussians
EEG	Electroencephalography
EKF	Extended Kalman Filter
FCM	Fuzzy C-Means
FEUP	<i>Faculdade de Engenharia da Universidade do Porto</i>
FFT	Fast Fourier Transform
fMRI	Functional Magnetic Resonance Imaging
FN	False Negative
FP	False Positive
GCV	Generalized Cross Validation
GNN	Global Nearest Neighbor
HMM	Hidden Markov Models
IBMC	<i>Instituto de Biologia Molecular e Celular</i>

ISBI	International Symposium in Biomedical Imaging
JPDFAF	Joint Probabilistic Data Association Filter
LMS	Least Mean Square Adaptive Filter
LoG	Laplace of Gaussian
MCMC	Markov Chain Monte Carlo
MEG	Magnetoencephalography
MHT	Multiple Hypothesis Tracking
MMSE	Minimum Mean-Squared Error
MRF	Markov Random Field
MSE	Mean Square Error
NNA	Nearest-Neighbor Association
NIT	Neuronal Intracellular Transport
NT	Neurotransmitter
PDAF	Probabilistic Data Association Filter
PDE	Partial Differential Equation
PDF	Probability Density Function
PET	Positron Emission Tomography
PNS	Peripheral Nervous System
PTV	Piccolo-Bassoon Transport Vesicles
RE	Recycling Endosomes
ROC	Receiver Operating Characteristic
ROI	Region Of Interest
SIFT	Scale-Invariant Feature Transform
SNR	Signal-to-noise Ratio
SOM	Self Organizing Map
STED	Stimulated Emission Depletion
SURE	Stein's Unbiased Risk Estimator
SV	Synaptic Vesicle
SVM	Support Vector Machines
SVP	Synaptic Vesicle Precursors
TN	True Negative
TP	True Positive
UI	User Interface
UKF	Unscented Kalman Filter
UPF	Unscented Particle Filter
WMF	Weighted Median Filter

# Chapter 1

## Introduction

Cells, the building blocks of life, contain a fundamental role in many human vital functions such as metabolism, immunity, and communication.

Neurons, specific type of cells composed of a cell body, dendrites and an axon, are responsible for processing the information in the brain. The communication between these cells is done by neurotransmitters, which migrate through the axon aggregated in vesicles. [1] The axonal transport of materials like vesicles is crucial for the survival and maintenance of neuronal network structures and physiological functions. [2] Recent studies confirm that the way these vesicles and other cellular constituents move along the axon differs when cells are in health or disease state, and can cause or be correlated with some human neurodegenerative conditions, including spastic paraplegia, Charcot Marie Tooth, Amyotrophic Lateral Sclerosis (ALS), Alzheimer's, Huntington's and Parkinson's. [3,4,5]

With the rapid growth of biotechnology and computer science it was possible to create different and advanced biomedical imaging techniques, allowing to extract, process and analyse a huge amount of data from microscopic images. This task improved our knowledge about many cellular and sub-cellular processes and in the future even better techniques are expected, coupled with smaller prices.

## 1.1 - Scope of the Thesis

Confocal microscopy is an optical microscopy technique widely used in the study of many vital processes such as the axonal transport in neurons.

Neuroscience researchers often cross a complex process to obtain the conclusions about the study, starting with a first phase in the lab, where the structures to analyse (vesicles in this case) are staining according to its biological/chemical properties.

The cell cultures are then placed in a confocal microscopy, which starts the process of acquisition, to obtain digital images. These images present some important features: the marked regions of interest (ROIs) with fluorescence, the background as black, the presence of noise and stains, and the axons path with less defined colour.

To extract the information from the image sequences acquired, image processing techniques, like denoising and enhancement, are combined to further obtain good segmentation and tracking results. The outputs are then analysed by the researchers to take proper conclusions about the vesicle movement. The complete schema is described in fig 1.1.

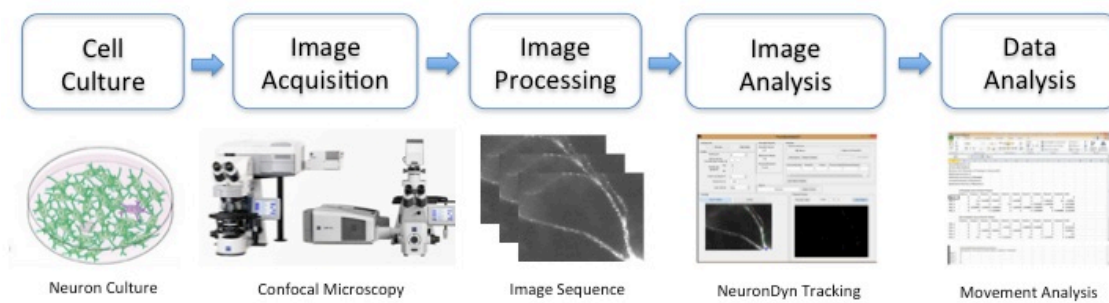


Figure 1.1 – Main steps of the vesicle's Neuroscience research.

Nowadays, biomedical imaging researchers have a large number of segmentation and tracking methods at their disposal, with different degrees of accuracy. After the segmentation of the image in its ROIs, according to specific features (e.g. centroid, area, boundaries, brightness, elongation), all objects found are linked between frames (tracking). In biological research, some dynamic phenomena found in cells and their constituents (birth, death, union, division, disappearance and clustering) represent serious difficulties to perform the association. After the tracking process, some meaningful quantitative measures related to the motility, diffusivity, velocity and morphology of the moving objects are obtained. [6]

Starting from this topic, this thesis presents a full analysis of the state of the art methods to implement segmentation and tracking processes, and a first software approach to give an interactive and precise solution to improve the researchers work.

## 1.2 - Motivations

### *Clinical Background*

Automatic tracking of the axonal transport is a challenging problem in today's neuroscience. In order to precisely evaluate the cellular traffic and prevent the referred problems, it's necessary to use technology in medical images at a very high temporal and spatial resolution, to be possible to detect and link the objects through the image sequence. [7] With the development of tracking algorithm capable of detect and follow the vesicles over the video, it is possible to remove the workload from researchers that make this same tracking manually, increasing the accuracy of the analysis.

The manual tracking faces many problems that can be solved with a semi or automatic tracking approach: subjectivity, since two researchers can obtain different results; errors in the analysis, connected to the huge amount of data and time performing the task (some videos can have hours, which results in long periods analysis); and the quality of the images, many times corrupted and blurred. These problems make it a time consuming and tedious task where, for example, the annotation of a video can take weeks or months. [8]

Although the satisfactory results obtained, the automatic tracking methods accuracy is very dependent of the application, leading to many errors in not-optimal scenarios. The absence of interaction with the investigator limits the automatic tracking performance (e.g., sometimes the researchers just need to analyse a specific set of frames, and not the whole video, that can lead to different conclusions). As researchers want to control the processing operation, semi-automatic tracking systems like *NeuronDyn* are developed with the need of inputs. The most common are: the adjustment of parameters, classification of true and false objects, classification of the detection in determined frames, selection of frames to the analysis, among others. [8]

The vesicle tracking problem addressed in this thesis is a specific case in object tracking, studied since some decades ago and with many applications. Both segmentation and tracking processes have to deal with many specific problems. In the first, touching and overlapping vesicles vary the area and number of detected objects, and background noise affects the segmentation accuracy. In tracking, low frame rates lead to abruptly changing position of objects, hindering its association. Vesicle collisions, merging and splitting and other dynamic properties also represent barriers to precise tracking.

### *Programming Approach*

To address the referred problem in the *NeuronDyn* approach *Matlab* programming was chosen. Despite *Python* offers a free and powerful language used in many biological applications, *Matlab* has some benefits: Extremely fast and concise coding, with a very complete set of functions in image analysis (saving time since there's no need to create them), helpful plotting tools, documentation and ability to call external libraries, and since it was implemented in other engineering courses.

### **1.3 - Main Objectives**

Starting from the initial need, the main objective of this thesis is to develop a robust algorithm for semi-automatic vesicle movement characterization in confocal microscopy images. This software will allow specialists to load videos of marked vesicles in neurons and to track them through time. Other possible option in *NeuronDyn* is to show the path of axonal transport where the vesicles are moving, and select only a determined region to analyse (between one start and end point of the path). Then, some vesicle's features are presented in the excel sheet, such as the direction of movement and velocity.

Although the algorithm is designed for vesicle movement characterization it can also be used to other cell structures like mitochondria or even cells. Indeed, in appendix A and B of this thesis the cell tracking topic is approached with two different solutions for cell tracking also developed for this thesis.

Due to the problems addressed, the algorithm developed should:

- Track simultaneously a large number of objects (vesicles);
- Allow quantifying some dynamic phenomena found in vesicles (birth, death, union, division, disappearance and clustering);
- Perform good denoising, enhancement and segmentation techniques;
- Be highly accurate.

Also, this thesis will presented a complete review and analysis of the state of the art in biomedical imaging processing, analysis and tracking, including semi and automatic tracking systems, which allow to process and quantify the information obtained by many imaging techniques. As other students will continue this project, this review will help them to select the proper methods to answer the researchers needs, allowing to improve the accuracy of the results. Also, this will give the reader a complete view of this topic, to understand the strategies applied.

## 1.4 - Publications

This Master thesis involved the following publications during its period:

[1] Carpinteiro, Frederico A., Costa, Pedro M., Espinoza, Mario Sáenz, Silva, Ivo M., Cunha, João P. S., Neurondynamics: A Method for Neurotransmitter Vesicle Movement Characterization in Neurons; In Proceedings of the IEEE International Symposium on Biomedical Imaging; April 2014 (poster presentation).

[2] Carpinteiro, Frederico A., Costa, Pedro M., Espinoza, Mario Sáenz, Silva, Ivo M., Cunha, João P. S., The Recursive Error Calculation Metric for Cellular Tracking; In Proceeding of the IEEE PGBiomed/ISC; July 2014 (poster presentation).

Also, some other publications are being prepared based in some of the contributions here presented.

## 1.5 - Work Structure

This thesis is organized in 7 main chapters, divided in two parts. In each chapter there is an introductory note with the objective of the chapter, specifying the problem and the solution presented.

Following this introduction, Part I (State of the Art) begins with Chapter 2, Neuronal Exchanges, where the properties of neurons and the neuronal intracellular transport are presented. Chapter 3, Neuroimaging Techniques, addresses the neuroimaging acquisition, explaining the most common techniques, especially confocal. Chapter 4, Image Processing and Analysis, describes the different image analysis and tracking methods and its applicability in the topic. This complete review presented in the State of the Art is mainly supported by recent papers and books, to obtain a current view of Neuroscience research.

Part II (*NeuronDyn*) addresses Chapter 5, where *NeuronDynamics* algorithm is described and the improvements are considered, Chapter 6, with the definition of *NeuronDyn*, and finally Chapter 7, which gives the conclusions of the present work. Appendix A and B present two important works about Cell Tracking also developed in this thesis.

The main part of this project took place in INESC TEC Research Institution, but it was also accomplished in IBMC Institute, in Porto.



# **Part I**

## **State of the Art**



# Chapter 2

## Neuronal Exchanges

This chapter will present the literature review associated with the biological phenomena in the transmission of information between neurons.

Starting from the Neuroscience view, neuron's properties will be studied, with its exchanges of information (neurotransmitters, vesicles and synapses), and relation with some common diseases.

This chapter will give the reader some basic information about the biological phenomena that will be studied in the *NeuronDyn* project, which allows researchers to take conclusions about the results.

### 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, structural, functional, computational and medical. [9]

Its studies started in the ancient Egypt, but only at the end of the 19<sup>th</sup> century, after the invention of the microscope, the nervous system was proved to being constituted by different neural cells. In the second half of the 20<sup>th</sup> century, associated with the advances in molecular biology and computational neuroscience, the scientific studies of the nervous system increased significantly, allowing to thoroughly study the brain. [10]

In the vertebrates, the nervous system is divided in the central nervous system (CNS, composed of brain and spinal cord) and the peripheral nervous system (PNS).

This nervous system is probably the most complex system that was studied over time, mainly due to the brain. [11]

## **2.2 - Neurons**

The human brain is composed by around one hundred billion neurons and one hundred trillion synapses. [12] There are thousands of distinguishable structures, connected in synaptic networks where the majority of the 20-25 000 genes of the human genome are expressed. One important feature of the human brain is its plasticity, changing the structure of synapses and their resulting functions through life. [13]

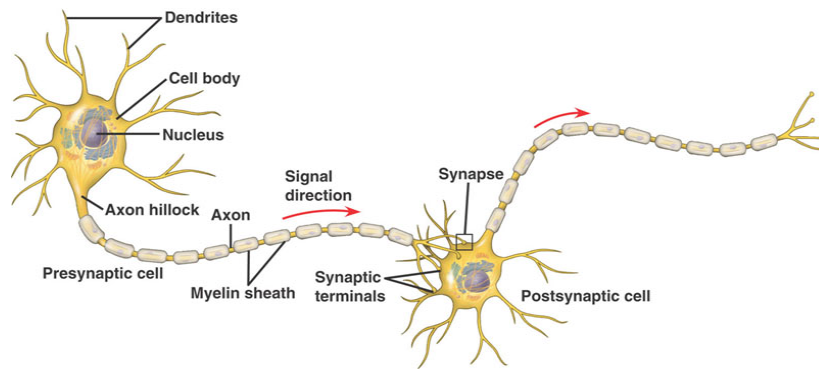
Neurons do not have cell division. In most cases, neurons are generated by special types of stem cells. Neurogenesis, the process by which neurons are generated from neural stem and progenitor cells, largely ceases during adulthood in humans. [14] But in two brain areas, the hippocampus and olfactory bulb, there are strong evidences for generation of substantial number of new neurons. [15,16]

Neurons, also called nerve cells, are highly specialized and electrically excitable cells that can detect, process and transmit electrical signals to other individual target cells, regulating the information in the brain. [1] Together with glial cells, that give them structural and metabolic support, they constitute the nervous system of almost all animals (with the exception of sponges and similar animals). [17]

In the vertebrates, most of the neurons are in the CNS, and a small part is in the peripheral ganglia, and sensory organs. They can be classified according to many parameters due to their diversity of functions performed in different parts of the nervous system, and wide variety in shape, size, and electrochemical properties. [18]

### **2.2.1 - Anatomy**

The neurons are normally composed by a cell body or soma, a dendritic arbor and an axon, being connected between them by synapses (figure 2.1). [1]



**Figure 2.1 - Representation of neuron's anatomy. [19]**

**Cell Body.** Cell body (soma) is the body of the neuron, which contains the nucleus and most of the protein synthesis and cytoplasmic organelles. The nucleus can range from 3 to 18 micrometers in diameter, while soma's diameter varies from 4 to 100 micrometers. [20]

**Dendrites.** Dendrites are thin projections that arise from the cell body, normally with hundreds of micrometers and multiple branches, forming the dendritic arbors. Usually, its function is associated to biological gates, with the reception of electrical signals. [18,21]

**Axon.** An axon, also known as nerve fibre, is a long and finer cellular extension that arises from the cell body in a place called axon hillock. [9] Its function is to conduct electrical and chemical information out of the neuron's cell body (anterograde transport) [22], but also back to it (retrograde transport). [23] Neurons can't have more than one axon, although some types don't have any, transmitting the signals directly from their dendrites. Axons can have length of 1 meter in humans and even more in other species, but unlike dendrites, they usually maintain the same diameter as it extends. [20] There are some different internal characteristics between axons and dendrites such as the regular presence of ribosomes and granular endoplasmic reticulum in dendrites, contrary to axon. [9]

### 2.2.2 - Electrical Potential

Neurons are electrically active due to a potential difference between the interior (negative) and the exterior (positive) of the cell. [24] This gradient is maintained through the action of sodium/potassium ATPase and unidirectional ion channels. With

the sodium/potassium ATPase, neurons contain high intracellular concentration of potassium but low intracellular concentration of sodium, that keep them below the threshold level for firing an action potential (AP). [25,26]

When the neuron is stimulated, there's a variation in the permeability in a section of the membrane, allowing sodium ions to cross into the neuron and altering the charge inside (to positive - depolarization). If this phenomenon reaches the threshold value, an AP begins, causing the following section of the membrane (along the axon) to become permeable, changing the charge too. In the previous section there's an outflow of potassium ions to the exterior of the neuron, allowing that section to be in rest state. The propagation of the AP occurs without distortion. [24,27]

The axon hillock is the part of the neuron with the greatest density of voltage-dependent sodium channels, being the most easily excited part of the neuron and the spike initiation zone of the axon, with the most negative AP threshold. This region of the neuron is very common to have connects to other neurons (synapses). [24]

### 2.2.3 - Synapses

The synapse is the structure that allows neurons to pass the electrical or chemical signal to another cell. On them, the plasma membrane of the signal-passing neuron (presynaptic) closely adjoins the membrane of the target cell (postsynaptic) and special molecular structures serve to transmit electrical or electrochemical signals across the gap. One single axon, with all its branches, can innervate multiple parts of the brain and generate thousands of synaptic terminals. [28,29]

Axonal synapses can be classified according to the type of postsynaptic structure, being the most common axodendritic synapses, where the content (neurotransmitters) is sent from the axon of one neuron to the dendrite of another. There are two different types of synapse natures:

- Electrical synapse, in which information is transmitted via direct flow of electrical current at gap junctions, a local where the membranes of the two cells are held together by paired channels in each membrane. Ions passively flow through these pores or channels from the presynaptic to postsynaptic channels thereby allowing the ionic current to influence the postsynaptic potential. In this transmission, due to the size of the gap junctions channels, can be possible a bidirectional flow of material. This synapse has faster transfer of signals between cells. [29,30]

- Chemical synapse, in which electrical activity of the presynaptic neuron is converted, through activation of voltage gated calcium channels, into the release of a chemical called a neurotransmitter in the synaptic cleft, that binds to receptors located in the postsynaptic cell. [29,30]

### 2.2.4 - Neurotransmitters

Neurotransmitters (NT) are endogenous chemicals like glutamate, acetylcholine and dopamine, which transmit signals between neurons in the synapses. They travel in the interior of membrane-bounded organelles called vesicles. They 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 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. [32,33]

## 2.3 - Neuronal Intracellular Transport

The neuronal intracellular transport (NIT) is an essential net to maintain the structure, function and survival of neurons, comprised of microtubule tracks with protein motors that shuttle intracellular contents bidirectionally in an anterograde (to the axon's terminal) and retrograde (to the soma) form. [2,34,35]

The microtubules are long thin tubes (with diameter about 25 nm) structures that provide a base for the movement of molecular motors throughout the neuron. They're formed of repeated units of a single globular protein (tubulin), which is in turn made up of two subunits of  $\alpha$  and  $\beta$  tubulin. Microtubules are dynamic and polarized, with a faster growing plus-end and slower growing minus-end. In the axon, the microtubule plus-end is oriented in an anterograde direction, whereas the minus-end is oriented in retrograde direction. In dendrites, microtubules may be oriented in either direction. [36,37] Molecular motors (proteins), attached to the microtubules, shuttle cargo material up and down the neuronal processes, using adenosine triphosphate (ATP) as fuel. [26]

**Anterograde transport.** The kinesin family proteins are responsible for anterograde (plus-end directed) transport. Kinesins are proteins that have an ATP-dependent motor that adheres to the microtubule on one end and have variable regions that provide some of the binding specificity to the transported cargo. When activated,

kinesin molecules transport membrane vesicles, synaptic vesicles and neurotransmitters at 200–400 mm/day (fast transport), and larger cargo such as microfilaments and neurofilaments at 0.1–10 mm/day (slow transport). [38,39] Slow transport uses the same mechanisms as fast transport however, due to multiple pauses along the way, the overall speed of transport is slower. [40] Mitochondria can move by both fast and slow transport but generally move somewhere in between the two at 50-100 mm/day. [41,42]

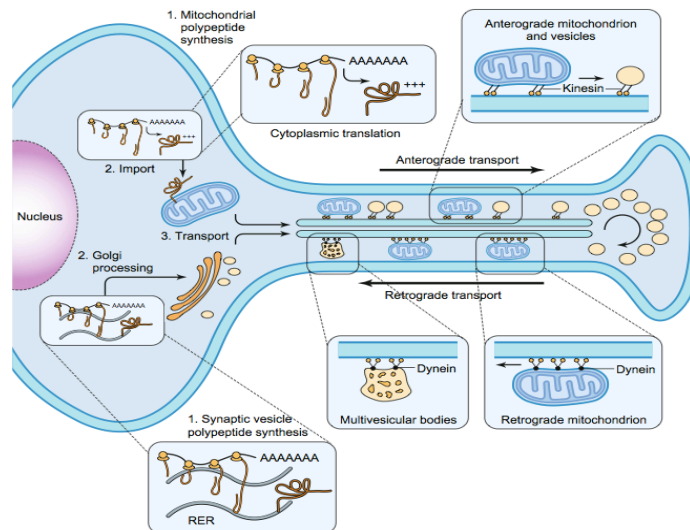
**Retrograde transport.** Retrograde transport is mediated by the dynein/dynactin complex, which shuttles materials toward the cell body at a rate of 100–200 mm/day. They have a key role in retrograde transport of neurotrophic factors, and provide signals to the cell body about distal cell damage and stress. [39,43,44]

Finally, for intracellular transport over shorter distances (e.g., at the end of the axonal structures), actin filaments with myosin motors transport materials to their final destination. [39,45]

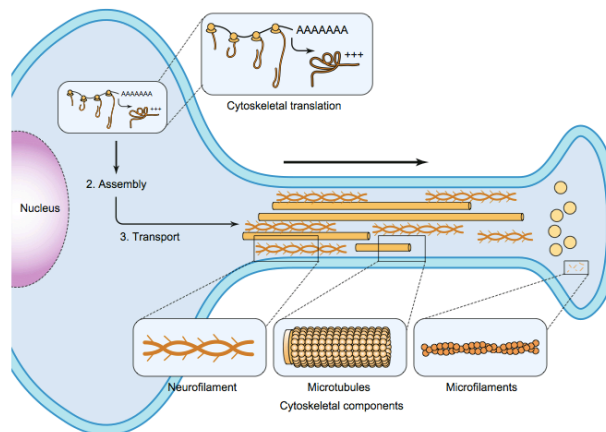
The summarized information of the axonal transport can be observed in the table 2.1, where the axonal transport types are indicated as well as the velocity/rate of transportation and carried substances. An overall view of both fast and slow axonal transport can be found in the figures 2.2 and 2.3.

Table 2.1 — Axonal Transport : types, velocity and carried substances. Adapted from [46].

Axonal Transport		Rate(mm/day)	Structures
Anterograde Transport	Fast Transport	200 - 400	Synaptic vesicles, enzymes, neurotransmitters, lipids
	Mitochondrial Transport	50 - 100	Mitochondria
	Slow Transport:		Tubulin, neurofilament proteins
	Slow Components a (SCa) Slow Components b (SCb)	0.1 – 1.0 2 - 6	Actin, clathrine, calmodulins, spectrin, cytoplasmic enzymes
Retrograde Transport		100 - 300	Prelysosomal vesicles, recycled proteins, HRP*, WGA**, neurotrophic viruses, Mitochondria.
*: Horseradish Peroxidase (HRP), **:Wheat Germ Agglutinin (WGA)			



**Figure 2.2 – Material moving in fast axonal transport, in both anterograde and retrograde directions. [39] The membrane-bound organelles are synthesized, attached to motor molecules and transported in the axon using axonal microtubules.**



**Figure 2.3 – Movement of cytoskeletal material in slow axonal transport, in anterograde direction. [39] The cytoskeletal polypeptides are translated on cytoplasmic polysomes and assembled into polymers before their transport along the axon, only in the anterograde direction.**

**Vesicles.** Neurons have several types of vesicles: Synaptic vesicles (SVs), that contain the classical neurotransmitters; the dense core vesicles (DCVs) that transport neuropeptides and guidance molecules; Synaptic vesicle precursors (SVPs) which transport synaptic proteins mainly during synaptogenesis; Piccolo-Bassoon transport vesicles (PTVs) transport most of the active zone proteins during early development; and the recycling endosomes (RE). These types can be observed in the figure 2.4. [47]

The regulated secretion of chemical signals, like neurotransmitters, is mainly achieved by two different types of vesicles: SVs, responsible for the release of neurotransmitters, a well-described process in the literature, and DCVs, that release the neuromodulatory cargo, including many neuropeptides and growth factors, an unfamiliar process in neuroscience. SVs are enriched in synapses and locally recycle in

response to activity. In contrast, DCVs are synthesized in the Golgi complex and transported from the soma to their release site. [47,48]

In the neurotransmission process, in addition to SV mediated release of neurotransmitters, monoamines, growth factors and a variety of neuropeptides are secreted through the fusion of DCVs. DCVs secrete several important neuropeptides that are involved in many different processes in the brain. About 90 genes that encode precursors of peptides have been identified as neuropeptides. [32,48,49]


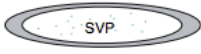
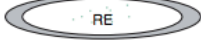
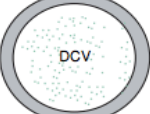
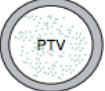
Vesicle type	Marker proteins	Cargo	Reference
 ~ 40 nm	SV2 syntaxin1 SNAP25 synaptotagmin1 synaptophysin Rab3	neurotransmitter	Zhang et. al., Neuron, 1998
 ~1 µm diameter	VAMP2 Rab3 PIP2 synaptotagmin1 synaptophysin VDCC amphiphysin	presynaptic components	Sytnyk et. al., J of Cell sci, 2004
 ~1 µm diameter	Rab11 transferrin	membrane components	Hsu et. al., Current op cell biol, 2010
 100 - 150 nm	Chromogranin Secretogranin	neuropeptides monoamines	Hsu et. al., Current op cell biol, 2010
 ~ 80nm	Bassoon Piccolo syntaxin SNAP25 N-Cadherin RIM munc13 munc18 VDCC	active zone proteins	Grace Zhai et. al., Neuron, 2001

Figure 2.4 – Neuronal types of vesicles , its marker proteins and cargo [47].

**Neuronal Intracellular Transport Alterations.** Vesicle transport in neurons is fundamentally more complex than in most other cell types because neurons tend to form long and elaborate processes, which can grow over great distances. Highlighting the importance of vesicle transport in neurons is the fact that deficits or disruption to vesicle transport, and other axonal transport organelles, may contribute to the development of several disease states. [4,5,50] Huntington's for example, is a disease characterised by the disruption of axonal and vesicle transport. This leads to deficits in cell signalling and eventually cell death. [51] The same can be seen in Alzheimer's disease, where alterations in axonal and vesicle transport are one of the earliest detectable changes that may contribute to disease progression. [52,53,54] This examples and others can be found in the table 2.2. This information shows that studying the physiological and pathophysiological mechanisms of vesicle transport is particularly important in neuroscience

Table 2.2 — Diseases related to the NIT [55].

NIT alteration	Disease	NIT dysfunction	Nervous system
Primary mutation of NIT component	Distal spinal and bulbar atrophy	Retrograde transport	PNS
	Perry syndrome	Retrograde transport	CNS
	Hereditary spastic paraplegia-10	Anterograde transport	CNS/PNS
	Charcot-Marie-Tooth 2A1	Anterograde transport	PNS
	TUBB3 syndrome (axonal sensorimotor peripheral neuropathy variant)	Microtubule function	CNS/PNS
Second dysfunction of NIT	Alzheimer disease	Anterograde transport	CNS
	Huntington disease	Anterograde and retrograde transport	CNS
	Parkinson disease	Anterograde transport	CNS
	Prion disease	Retrograde transport	CNS
	Hereditary spastic paraplegia 4	Microtubule function	CNS/PNS
	Amyotrophic lateral sclerosis	Anterograde and retrograde transport	CNS/PNS
	Giant axonal neuropathy	Intermediate filament transport	CNS/PNS
	Colchicine and vincristine peripheral neuropathy	Microtubule function	PNS
	Hexacarbon peripheral neuropathy	Anterograde and retrograde transport	PNS
	Other Charcot-Marie-Tooth forms	Anterograde, Retrograde and Intermediate filament transport	PNS



# Chapter 3

## Neuroimaging Techniques

This chapter will present the Image Acquisition process in the Neuroscience research.

Starting with an overview of the Neuroimaging, optical and electron microscopy techniques are then compared, and the different confocal methods are presented.

This chapter will give the reader the basics of the conditions in which the neuronal images are taken, the associated problems and the limitations.

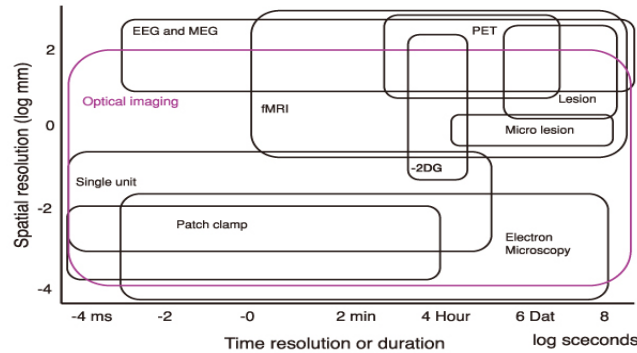
### 3.1 - Neuroimaging

Neuroimaging, a neuroscience discipline, can be divided in two categories: Structural Imaging, which deals with the diagnosis of injuries like tumours (e.g. by showing contrast between structures); and Functional Imaging, used to indirectly measure brain functions (used e.g. in metabolic diseases, neurological and cognitive psychology research and to build brain-computer interfaces).

The automation of the acquisition and interpretation of biological data in microscopy began in the 1950s. This first step was concerned about 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 broad spatial and temporal scales. [56]

There are different neuroimaging techniques like functional magnetic resonance imaging (fMRI), positron emission tomography (PET), magnetoencephalography (MEG) and Light (optical) imaging. Other techniques such as EEG

(electroencephalography), single-unit recording, patch clamp, Brain lesions and micro lesions, Electron Microscopy and 2-Deoxy-D-glucose (2DG) are also used. In figure 3.1 there is a comparison of the spatial and temporal resolution of some neuroimaging techniques, that shows the emphasis of the optical imaging methods. [56, 57]

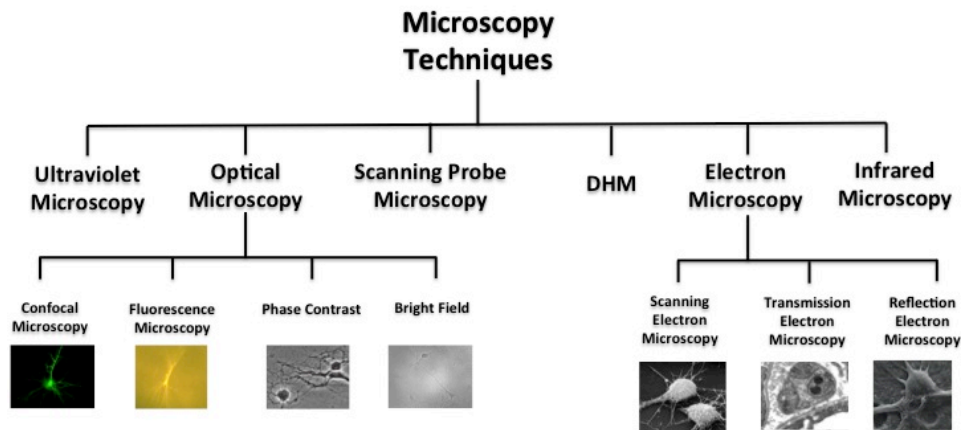


**Figure 3.1** – Comparison of the spatial and temporal resolution and scope of some imaging techniques used in Neuroscience [56].

### 3.2 - Optical vs Electron Microscopy

Both Electron and Light (Optical) microscopy are used to visualize very small structures in biology and materials sciences, but with very different methods as the first use electrons and the second photons (light rays) for visualization. Although electron microscopy has higher resolution due to the physical limitations of the wavelength of the light, the light microscopy is a much novel technique that has lower price, simplicity of using, and possibility to observe moving objects (like living cells) and colour. On the other hand, electrons do not possess colour, so the image is black and white in electron microscopy. Thus, optical microscopy is used to observe a lot of biological activities, such as cell division or vesicle movement. Figure 3.2 shows the main techniques in microscopy field, with more focus on optical and electron microscopy.

Optical imaging includes illumination and contrast methods (e.g. Differential interference contrast –DIC-, bright field microscopy), Fluorescence methods (fluorescence microscopy, confocal microscopy, multiphoton microscopy, image deconvolution) and sub-diffraction limit techniques (diffraction limit, Stimulated emission depletion – STED). [58]



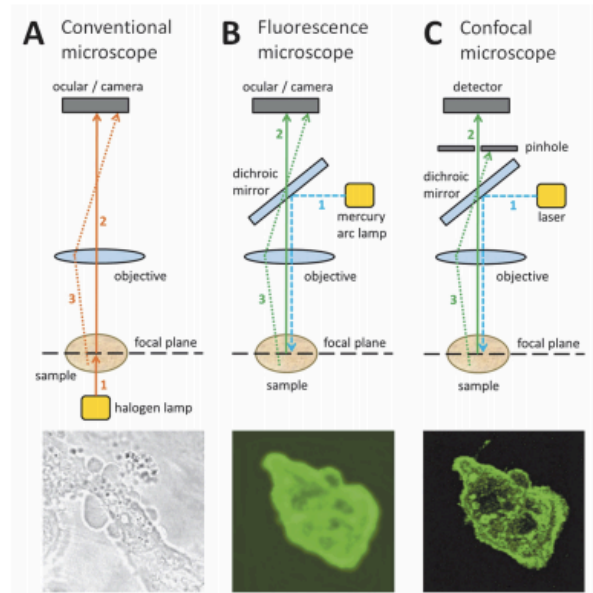
**Figure 3.2** - Main Techniques in Microscopy field.

Although the success of fluorescence microscopy, the capture of dynamic phenomena in live cells is limited due to phototoxicity (cells that contain fluorescence are exposed to illumination, generating chemical species like oxygen radicals that damage subcellular components and affect cell behaviour and fate) and photobleaching (loss of fluorophores ability to fluoresce during time-lapse, which limits the number of image acquisitions). Microscopes want to assure high image acquisition speeds of fluorescent proteins and synthetic dyes in live cells with reasonable contrast and minimal photobleaching. This means they must be able to quickly scan the field of view and record data using detectors with high quantum efficiency, making confocal microscopy is a good solution. [8]

## 3.3 - Confocal Microscopy

Confocal microscopy is an optical imaging technique used to increase optical resolution and contrast of a micrograph that provided important advances in biomedical imaging. It offers several advantages over other techniques, including the ability to control depth of field, elimination or reduction of background information from the focal plane, and the capability to collect serial optical sections from thick specimens. Confocal uses point-by-point illumination to create the images and a pinhole in front of the detector to reject out-of-focus light. The laser light source has very high intensity and the emitted photons reflect in a dichroic mirror, which directs it to an assembly of vertically and horizontally scanning mirrors. These motor-driven mirrors scan the laser across the specimen and the optics remain stationary. The specimen is axially illuminated everywhere (without different angles) avoiding optical aberrations. Confocal imaging creates extremely high-quality 2D sharp images, and allows collection of data in three

dimensions. This optical imaging technique is very used in cell biology that relies on imaging both fixed and living cells and tissues. Figure 3.3 presents a comparison between conventional, fluorescence and confocal microscope. [59]



**Figure 3.3** - Comparison between a conventional, fluorescence and confocal microscope. In A, the light goes through the sample, and the resulting light from it passes directly through the ocular, also with light from above or below the focal plane. In a widefield fluorescence microscopy (B), excitation light is reflected by a dichroic mirror to the sample, and emission light from the sample passes through the dichroic mirror to the ocular, also with light from above or below the focal plane. Confocal scanning confocal microscope uses similar approach to fluorescence microscope, but with a pinhole in front of the detector, excluding most of the out-of-focus light. [59]

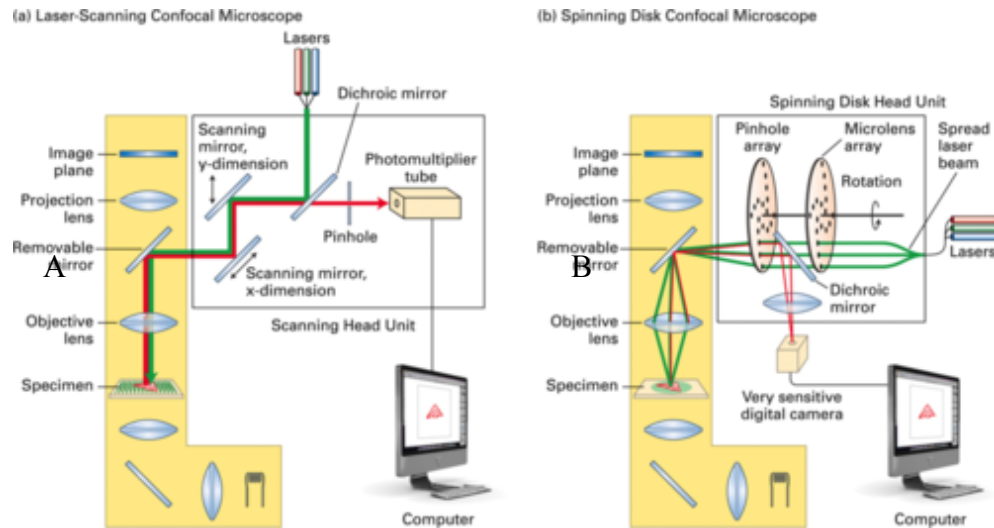
The two principal confocal systems are Confocal Laser Scanning Microscopy (CLSM) and Spinning Disk Confocal Microscopy (fig. 3.4).

### 3.3.1 – Laser Scanning Confocal

Laser Scanning Confocal microscope focus a single beam on the specimen plane to sequentially point-scan a ROI with spatial filtration of the emission light through a single pinhole that rejects light from out-of-focus regions. It represents a slow process of acquisition but with good flexibility, presenting unbeatable results in the acquisition of single images. [60]

### 3.3.2 – Spinning Disk Confocal

The Spinning Disk method contains an array of pinholes and microlenses, scanning the specimen with many light points. Spinning Disk presents a much faster technique, collecting for example 2000 images per second. Also presents a good bleaching response but has poor flexibility, which make it better to obtain movies.



**Figure 3.4** - Differences between a) CLSM and b) Spinning Disk Confocal Microscopy techniques. [61]

The image information is gathered point by point with the photomultiplier tube detector, and then digitized for processing by the computer, which also controls the scanning mirrors and other devices to collect and display the images. After the acquisition and storage of the digital data, the size of the images varies from few to several hundreds of megabytes that are usually related to the resolution of the camera, typically varying from 512 x 512 to 1392 x 1040 pixels. This represents a huge amount of data to be analysed manually. This process creates a need of robust and automatic or semi-automatic image processing tools.



## **Chapter 4**

# **Neuroimage Processing and Analysis**

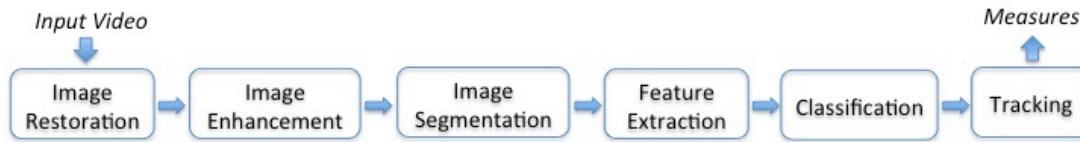
This chapter will address the Image Processing, Analysis and Tracking of the information.

After some basic definitions about images, the most important denoising, enhancement and segmentation methods are explained. Then, the feature extraction is associated with clustering and classification techniques, finishing with the different tracking methods and the evaluation of the CAD Systems.

This chapter is essential to implement computer vision methods in the present research. As other student will address this study in the future, the review here presented tries to be as more complete as possible.

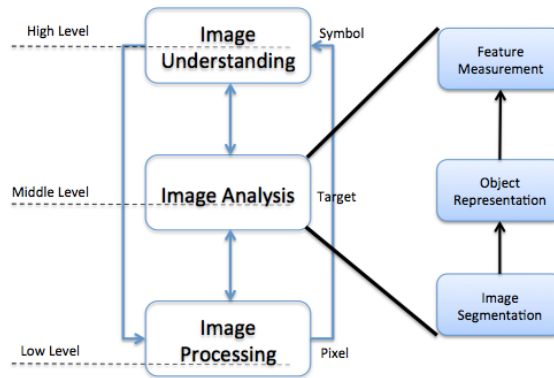
### **4.1 - CAD System**

Nowadays, it's extremely important to extract, analyse and represent information from digital images obtained by neuroimaging techniques. These image-processing tools are widely used in many engineering fields like biomedical, computer science and metallurgy. [62] The typical tasks of microscopy image processing are detecting and segmenting boundaries of objects, classifying and recovering the motion of the moving objects over time. [63] In this section each step of a computer aided-diagnosis (CAD) system (fig. 4.1) will be approached with the main techniques to achieve it.



**Figure 4.1** – Typical workflow of a tracking CAD system.

The three main phases of image engineering treatment are image processing, analysis and understanding, which have different operational levels. Image Processing has relatively low-level operations, mainly on the pixel-level. Image Analysis enters the middle-level, and focus on measuring, expression and description of targets. Finally, Image Understanding is mainly high-level operation, focussing on the operation and illation of data symbol, which abstracts from the description. Figure 4.2 resumes the image treatment in the engineering field, with focus on the Image Analysis, the core process.



**Figure 4.2** - Image Engineering main phases. Adapted from [64]

## 4.2 - Image Basics

An image is a 2D function  $I(x,y)$  where  $x$  and  $y$  are spatial coordinates and  $I$  is the intensity for each pair of coordinates  $(x,y)$  that represent a pixel. There are four types of images: RGB - With three matrices (3 channels), one for each color, red, green and blue, all normalized between  $[0,1]$ ; Binary images - pixels have 0 (black) or 1 (white) values; Grayscale images – with one matrix where values represent the intensity in each point. Normally with values from 0 to 255 or from 0 to 1, representing shades of gray. Indexed Images - two matrices, one with the index to the second one composed with the quantities of R, G and B for each pixel.

**Feature Detection.** Feature detection is defined as locating pixels in the image that have some distinctive characteristics. The most simple image features are: Edges - points that form a boundary between two regions (high changes in gray-level) and have strong gradient magnitude; Corners - points that have rapid changes in direction; Blobs - containing points with a local maximum of an operator response; and Ridges - generalizations of a medial axis for elongated objects (used for road extraction in aerial images and for blood vessels in medical images).

Many features are used for different tasks in biological images: area, perimeter, circularity, compactness, shape, size, location, convexity (convex hull), concavity, smoothness, variance, average, gradient norm, radial gradient, roundness, standard deviation, eccentricity, major and minor axes, roughness, moments, among others.

The gradient is a mathematical operator vector for each pixel and its components are calculated by the derivatives of the image in the direction of the coordinate axes. The gradient of an image  $I(x,y)$  at the location  $(x,y)$  is defined by the vector in the equation 1, and the magnitude and direction of the gradient are represented in equations 2 and 3 respectively.

$$\nabla I = \frac{\partial I}{\partial x} + \frac{\partial I}{\partial y} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} \quad (1)$$

$$\text{mag}(\nabla I) = [G_x^2 + G_y^2]^{1/2} \quad (2)$$

$$\text{angle}(\nabla I) = \arctan \left( \frac{G_y}{G_x} \right) \quad (3)$$

$G_x$  and  $G_y$  represent the partial derivatives of the image with respect to the horizontal and vertical direction. The gradient is computed by convolving two spatial filters (or mask), which implement first order derivatives, with the image in the horizontal and vertical directions to get  $G_x$  (horizontal edge image) and  $G_y$  (vertical edge image). The most common masks are Roberts and Sobel. [65]

### 4.3 - Image Restoration/Denoising

The quality of the acquired images is still low these days, which is normally associated to many artefacts that affect its processing, like noise, blur, uneven illumination, optical aberrations or signal inhomogeneities. Different devices and acquisition procedures can lead to different noise models, like Gaussian or Poisson. They are also related with inappropriate acquisition parameters (exposure time, binning, gain). This first step of image processing involves the manipulation of the data to improve a visually quality to

the original image by removing/reducing the events incurred while the image was obtained. [66]

#### 4.3.1 – Noise Models

All medical images contain some noise. Noise is composed of random values that follow a distribution which is expressed by a probability density function (PDF). The most known noise models, represented in table 4.1, are studied:

- Gaussian Noise (Additive White Gaussian Noise), which represents statistical noise having PDF equal to the normal distribution (also known as Gaussian distribution). The difficulty of suppressing it is high since it corrupts almost all pixels in the image;

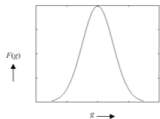

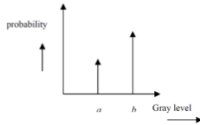
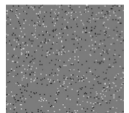
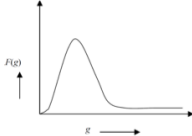

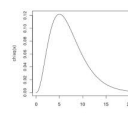

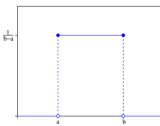
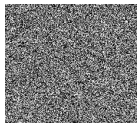
- Impulsive or Salt-and-pepper noise. This kind of noise is usually associated with defective camera pixels, analog-to-digital converters and transmitted bit errors. Images have dark pixels in bright regions and bright pixels in dark regions;

- Speckle noise – multiplicative noise that occurs in many imaging systems. The source of this noise is attributed to random interference between the coherent returns and follows a gamma distribution.

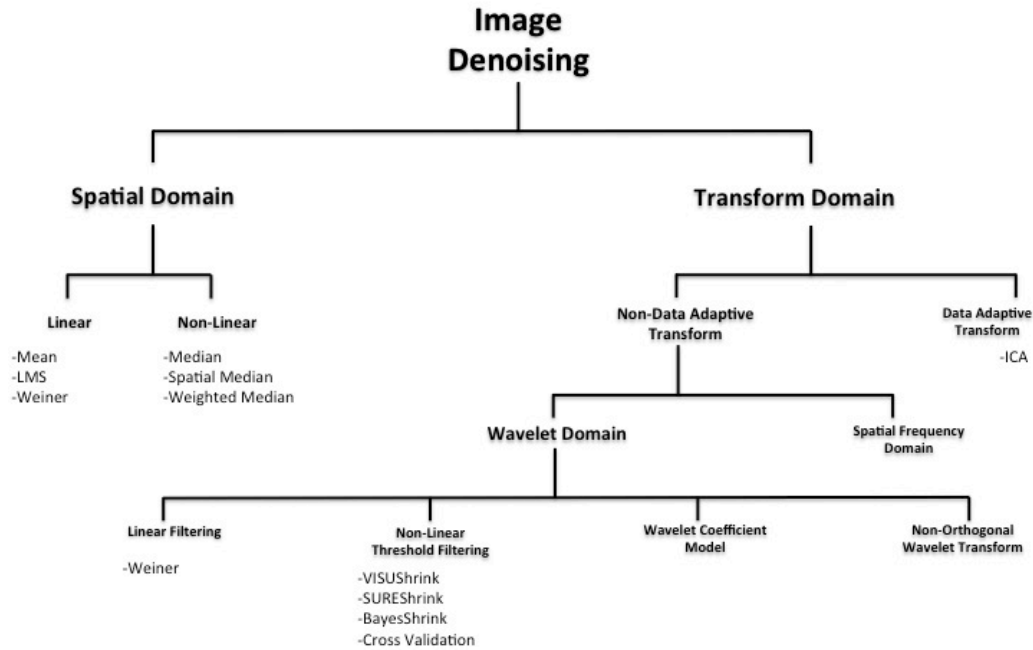
- Photon shot noise, which is caused by statistical fluctuations in the observed signal following the Poisson distribution. This is the most common type in fluorescence and confocal microscopy, where the image is created by the accumulation of photons over the detector;

- Quantization noise (uniform noise) – This noise follows an approximately uniform distribution, and is caused by quantizing the pixels of a image to a number of discrete levels. [67]

Table 4.1 – Main types of noise in images, its distribution formula, representation and an example.

Type of noise (distribution)	Formula	Representation of distribution	Image Example
<b>Gaussian Noise</b> <b>(Gaussian)</b>	$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2 / 2\sigma^2}$		
<b>Impulsive Noise</b> <b>(Salt-and-pepper)</b>	$\begin{aligned} p(z) &= P_a & \text{for } z = a \\ p(z) &= P_b & \text{for } z = b \\ p(z) &= 0 & \text{otherwise} \end{aligned}$		
<b>Speckle Noise</b> <b>(Gamma)</b>	$\begin{aligned} p(z) &= \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ p(z) &= 0 & \text{for } z < 0 \end{aligned}$		
<b>Shot Noise</b> <b>(Poisson)</b>	$P(z) = \frac{\lambda^z e^{-\lambda}}{z!}$		
<b>Quantization</b> <b>(Uniform)</b>	$\begin{aligned} p(z) &= \frac{1}{b-a} & \text{if } a \leq z \leq b \\ p(z) &= 0 & \text{otherwise} \end{aligned}$		

Current denoising techniques combine suitable filter and statistical estimation. Typically, the image is transformed, using filters, into a domain where the noise can be identified and statistical estimation is then performed to remove the influence of noise. There are several methods to restore an image from noisy distortions. Selecting and implementing the appropriate ones to the image remains the major difficulty once denoising methods tend to be problem specific. The main techniques can be dividing according to figure 4.3:



**Figure 4.3** - Image Denoising Techniques Classification. Adapted from [66]

### 4.3.2 – Spatial Domain

The application of spatial filters can be divided in Linear filters and Non-linear filters. In the first, the output values are linear function of the pixels in the original image and are easier to analyse mathematically. The non-linear present more accurate results from their ability to reduce noise levels without blurring the edges.

**Linear Filters.** These filters are good in the presence of Gaussian noise, can remove noise to a reasonable extent and are easy and fast to implement but have a cost of blurring edges. The Mean Filter calculates the average value of the image in a predefined area and the central pixel intensity value is then changed to the average value of the pixels in the neighbourhood. It smoothes the image by reducing the intensity variation between adjacent pixels being a good solution to Gaussian noise. In the Least Mean Square Adaptive Filter (LMS) the weight matrix varies after each iteration (unlike mean filter). Adaptive filters are capable of denoising non-stationary images, which have abrupt changes in intensity. 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 requires the information about the noise spectra and achieves optimal denoising for stationary signals.

**Non-linear filters.** Non-linear filters are applied when noise isn't additive. In Median filtering the entire pixel values of the mask are sorted into a numerical order and then the median value is considered. This filter has better response to the edges than the mean filter. The Spatial Median Filter is obtained by calculating the spatial depth between a point and a set of points. After finding the spatial depth of each point lying within the filtering mask, this information is used to decide whether the central pixel of window is corrupted or not. Then, it finds the spatial depth of each pixel within the mask and sorts them in descending order, selecting the largest spatial depth to represent the median of the set. Finally, the Weighted Median Filter (WMF) gives more weight to some values within the window. [66, 68]

### 4.3.3 – Transform Domain

**Spatial Frequency Filtering.** These methods apply low pass filters using the fast Fourier Transform (FFT), which maps a signal into its component frequencies. In the frequency smoothing, the removal of noise is achieved by designing a frequency domain filter and defining a cut-off frequency when the noise components are decorrelated from the useful signal. But these methods are time consuming and dependent of the cut-off frequency and filter function behaviour.

**Wavelet Domain Filtering.** Over the past decade, wavelet transform has received a lot of attention from researchers in many different areas. Unlike Fourier transform, which has good frequency localization but destroys the spatial information, the wavelet transform stabilises a representation of data on different scales preserving the spatial/time organization. Linear filters such as Wiener filter in the wavelet domain yield optimal results when the signal corruption can be modelled as a Gaussian process and the accuracy criterion is the mean square error. But the most investigated domain in denoising using wavelet transform is the non-linear coefficient thresholding based methods. One of the most known algorithms in non-adaptive threshold is VisuShrink, which follows a hard thresholding rule and uses a threshold value  $t$  proportional to the standard deviation of the noise. Adaptive Threshold methods like SureShrink use a hybrid of the universal threshold and the Stein's Unbiased Risk Estimator (SURE) threshold, performing better than VisuShrink. BayesShrink minimizes the Bayes' Risk Estimator function assuming Generalized Gaussian prior and thus yielding data adaptive threshold (BayesShrink outperforms SureShrink most of the times). Cross validation replaces wavelet coefficient with the weighted average of neighborhood coefficients to minimize generalized cross validation (GCV) function

providing optimum threshold for every coefficient. The output from BayesShrink method is normally closer to the high quality image when compared with SureShrink and VisuShrink. [66]

## 4.4 - Image Enhancement

Image Enhancement is used to improve the contrast, edges and visibility of the characteristics of interest, without causing distortion or artefacts. [69] To accomplish this process, several methods can be used, which are divided in two different domains, Spatial and Frequency domains. Figure 4.4 presents a classification of the Image Enhancement.

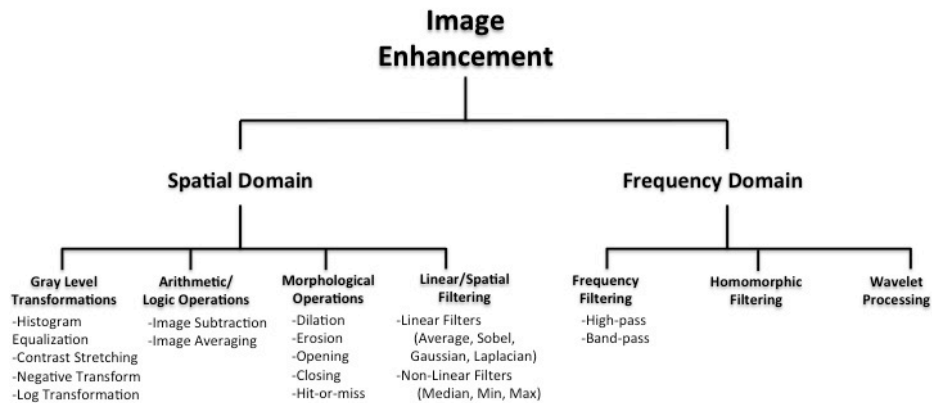


Figure 4.4 - Image Enhancement methods.

### 4.4.1 –Spatial Domain

In the Spatial Domain images are processed directly using pixel-by-pixel transformations or using the neighbourhood to manipulate the pixel's intensity. The most important methods are:

- Gray Level Transformations - Histogram-based approaches using contrast stretching (linear expansion), histogram equalization or negative transformation;
- Arithmetic/Logic Operations - Image Subtraction (to detect dynamic phenomena), Image Averaging, Addition, Multiplication or Division and also logic operations (AND, OR, NOT);
- Morphological Operations - Erosion, dilation, opening, closing or hit-and-miss algorithm;

- Linear Filtering/ Spatial Convolution - The transformations are applied taking in consideration the neighbourhood by convoluted the image with a mask, a group of weights to modify the pixel's value. These filters can be linear such as sum or difference based filters; first order derivatives (use gradient and directional derivatives) filters like Sobel, Gaussian, Prewitt or Roberts, which produce edge enhancement; and second order derivatives (laplacian) filters. Non-linear filters are, for example, median, maximum or minimum filters.

### **4.4.2 – Frequency Domain**

In the Frequency Domain, the Fourier transform is applied and the operation is based in the characteristics of frequency of the image. [69] Frequency filtering uses high-pass, low-pass and pass-band filters to obtain specific frequency components. Low frequencies represent the main structures of the image, while high frequencies give information about the shape details, edges and noise. In many applications it's important to enhance the reflectance (how objects reflect the light) and reduce the brightness of the illumination (light component), achieved by using the Homomorphic filtering.

One important task in biomedical applications is the enhancement of tube-like structures like the dendrites of neuron cells. Most algorithms assume that these structures have an elliptical or semi-elliptical shape, although they have very irregularly shapes, compromising the algorithm's performance for detection.

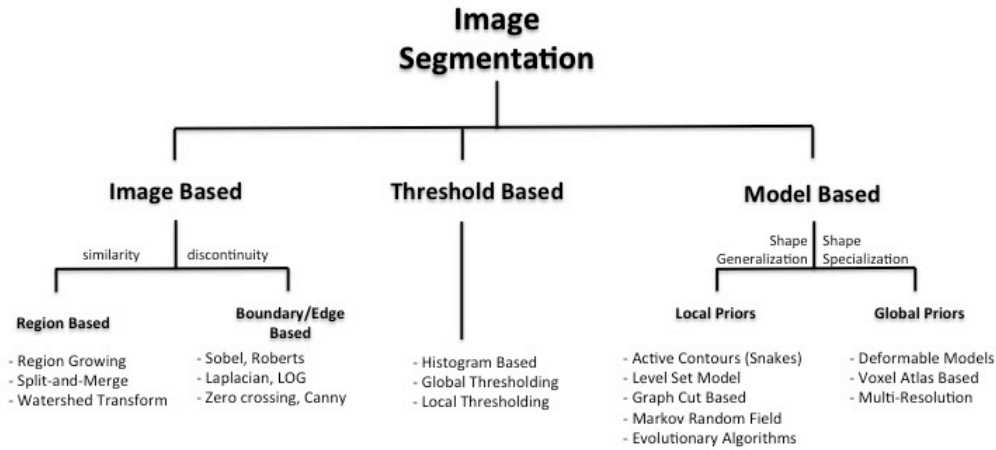
## **4.5 - Image Segmentation**

Image Segmentation is defined as the process of separating the image in its constituent's objects/regions. Based on the response of filters in the enhancement step, and according to exclusion criteria, that can be a property like grey level intensity, colour or texture, it's possible to map the object candidates (regions of interest – ROI) from the background.

Using microscopic images, our goal is to extract the cellular components of the images, which can have large variations of features (size, shape, orientation or texture). The major challenges in this field are connected to the segmentation of touching and overlapping cells, in low quality images. There are several reliable methods to do it, although there is a trade-off between speed, complexity and segmentation accuracy.

Taking in consideration the abundance of literature in image

segmentation, a complete overview is presented to a correct selection of the methods for neurons specific applications. Also, a new categorisation of the Image Segmentation for this topic is presented in figure 4.5 [71]



**Figure 4.5** – Image Segmentation overview.

The segmentation process can be categorized based in three characteristics: object boundaries, object homogeneity and shape. Boundaries and Homogeneity are image or signal based characteristics and are affected by image specific disturbances like noise or reconstruction artefacts. The thresholding techniques are in the middle of the Image and Model approaches as, in this field, they can be studied as object/background models (uniform background with irregularly placed objects) and considering that they use image information. Machine Learning methods are not considered segmentation techniques in this study but, however, they support segmentation, finding appropriate parameters. Finally, hybrid methods incorporate parts of the different techniques mentioned above.

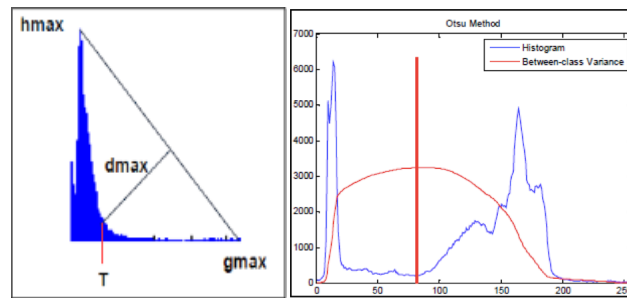
#### 4.5.1 – Thresholding Methods

The thresholding methods are the most used and simplest segmentation methods. They turn a gray-scale image into a binary image by defining a threshold value,  $T$ . This technique assumes that the image will be defined only in two classes of pixels: background and object pixels, each one with a different gray level.

There are some methods to find the threshold value. The most common is histogram visualization: With a bimodal histogram, two classes of values are well defined (background and objects) and the threshold value should be in its valley (mode method). Using Otsu method, its possible to find  $T$ , where between-class (interclass) variance is maximized and intraclass variance is minimized. With only one mode, the

triangle algorithm can be applied by calculating the maximum of the histogram and the maximum intensity of the image, computing a line between those points. Then, the perpendicular point with the maximum distance to the line is selected as threshold value. Figure 4.6 presents an illustration of the referred histogram-based methods. [72]

Gaussian filtering based on the Bayes decision theory is also used. The two component densities are extracted (object and background) and the mean ( $\mu$ ) and standard deviation ( $\sigma$ ) are computed, the parameters of the mixture density are estimated and a pixel with gray level is assigned as threshold.



**Figure 4.6** - Histogram based thresholding methods. A) Triangle method in a unimodal histogram, B) Otsu method applied to a bimodal histogram. [65]

As global threshold methods are very sensitive to parameters like ambient illumination and noise level, local methods can be used. The adaptive/local threshold methods, like Chow Kaneko, divide the image in regions and compute the local histogram and  $t$  for each region, grouping them to form the final image.

Finding the pixels that present a great response to the Laplacian filter, which normally present a middle intensity between the objects and the background, is another good way to find the threshold value. If there is no sufficient contrast between objects and background, the difficulty of the images and its histograms is higher. [73]

### 4.5.2 – Image Based

In the Image domain there are two main types of methods, the Region-based, where regions are segmented in pixels with similar properties, and the Edge-based, where segmentation is achieved in pixels with different properties.

#### Edge-based Algorithms

Edges and contours are formed by the intersection of two regions where there are abrupt changes (discontinuities) in the gray-level intensity. To obtain them sensible operators are used to identify the abrupt changes and to suppress the constant gray-level areas.

First derivative filters (like Sobel, Roberts and Prewitt) produce images with thicker edges while second derivative filters (like Laplacian, Laplace of Gaussian – LOG -, Canny) have better response to finer detail. LOG filter reduces the effect of noise by smoothing the image with a Gaussian filter and then using the Laplace operator. Canny's detection is the most complete since it has a combination of enhancement and detection. These methods are particularly good in images with high contrast between regions, contrarily to noisy images or with non-defined edges.

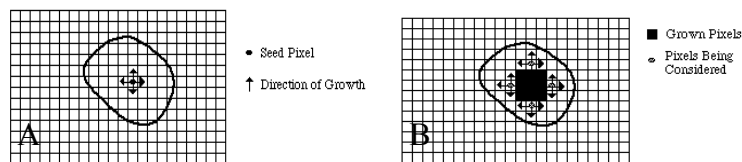
Occasionally detected edges have incomplete borders and line detection algorithms are used to solve it, with line or shape fitting approaches. With Hough Transform method an analytical representation of a line in 2D images is used to find the best fit to the edge by employing an accumulator as line detector. [74]

### Region-based Algorithms

Region-based methods group pixels into homogeneous regions according to a predefined criterion of connectivity. They are more resistant to noise than edge detection approaches but they also require higher computational cost and memory. Its applications are connected to many biomedical imaging processes, pixel aggregation and 3D reconstruction of shape.

The most common methods are the region growing, the split and merge and the watershed algorithm. Normally those methods face some problems like under or over segmentation, and to avoid it the selected criteria should be precise and some edge information can be added.

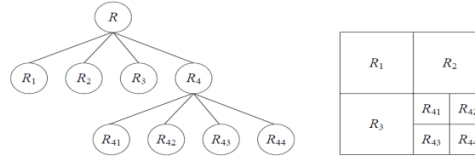
**Region Growing.** Starting from a group of seed pixels, the region grows through the annexation of neighbour pixels to the seed ones (fig. 4.7), with similar characteristics (intensity, texture, colour). The region stops to grow when the established criteria are achieved and no more pixels can satisfy it. [75] In situations like vesicle touching or clusters the seed points are not easy to be obtained, making this approach inappropriate.



**Figure 4.7** - Region Growing Segmentation method. a) Seed pixel and b) growing process. [76]

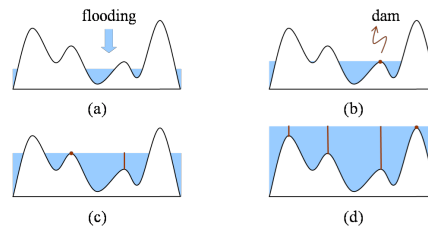
**Split and Merge.** Based on quadtree partition of the image, this method has a combination of splits and merges using the advantages of both. The whole image is considered as one region, being splitted in a recursively process into four quadrants

each time, until all sections satisfy the homogeneity criteria (fig. 4.8). Then, adjacent regions with similar attributes are merged following the same, or not, criteria. By taking into account edge information in the merge process it's possible to solve over-segmentation problems.



**Figure 4.8** - Split and Merge Segmentation method. [77]

**Watershed.** Contrary to region growing that works considering a neighbour layer, the watershed algorithm considers an intensity layer. After obtaining the gradient image, which is considered as a topographical surface, a flooding process starts from the minimum grey level. When flooding across two zones merge, a dam is built to identify that boundary and the process continues until it reaches the highest level on the landscape (fig. 4.9). This method is also a good solution to segment touching particles like vesicles, a common problem in biomedical imaging.



**Figure 4.9** - Watershed method. [78]

### 4.5.3 – Model Based

Model based methods are related to object's shape, which are image independent and in most cases also independent from acquisition techniques. These approaches assume that objects are present in a certain pattern in the images and model their shape using local or global priors.

#### *Local Priors*

The most known methods for image segmentation based on local shape priors use the partial differential equation (PDE), like active contours or snakes, but there are also graph based methods and Markov random field methods, among others.

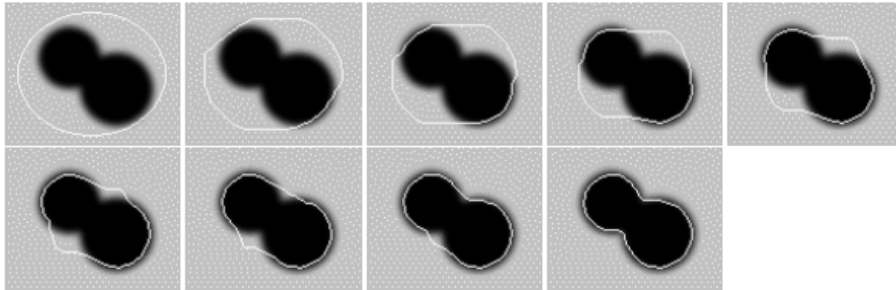
The basic concept of the PDE methods is to characterize a given

problem as an energy functional which satisfies a time dependent PDE. This process is mainly obtained by active contour models or snakes and level set.

**Snakes.** Active contours or snakes are computer-generated curves that move within the image to find object boundaries under the influence of internal and external forces. It's a iterative process where the shape of the snake is given by minimizing the equation 4:

$$S(s) = I(s) + E(s) \quad (4)$$

Where  $I(s)$  represent the internal snake energy and  $E(s)$  the external snake energy. The internal energy defines the rigidity and the tension of the contour and the external determines what kinds of image features attract the contour. Figure 4.10 presents an example of a snake application with optimization between iterations. The three dimensional generalization of snakes is called deformable model or deformable surface. A disadvantage of snakes is their dependency on initialization and lack of topological adaptation. [79]



**Figure 4.10** - Example of snake application, showing the stages between iterations. [79]

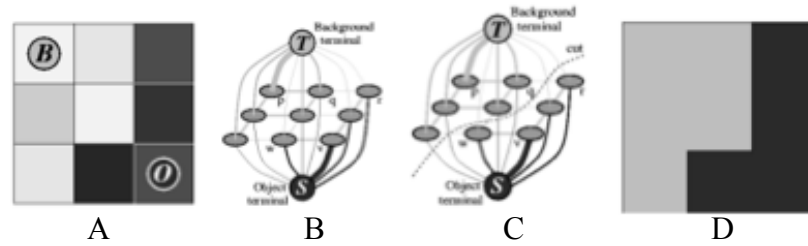
**Level Set Model.** Represent the evolving contour using a signed function (where its zero level corresponds to the actual contour). Then, according to the motion equation of the contour, it's possible to derive a similar flow for the implicit surface that, when applied to the zero-level, will reflect the propagation of the contour. Level set method is a powerful method for medical segmentation and can handle any of the cavities, concavities, convolution, splitting, or merging phenomena. This method needs to specify initial curves and if they are placed near symmetrically with respect to the object boundary, give good outcomes. [71]

**Graph cut-based approaches.** Based on combinatorial optimization, graph cut approaches the segmentation problem by minimizing an energy function defined on a

combination of both region and boundary terms. The image is modelled as a weighted, undirected graph. Usually a pixel or a group of pixels are associated with nodes and edges weights define the similarity or not between the neighborhood pixels. The graph (image) is then partitioned according to a criterion designed to model efficiently the clusters (Fig. 4.11). Each partition of the nodes output from these algorithms is considered an object in the image. A cut is a subset of edges by which the graph will be partitioned into two disjoint sets  $A$  and  $B$  and the cut value is usually defined using equation 5:

$$\text{cut}(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (5)$$

where  $u$  and  $v$  refer to the vertices in the two different components. The cost function is defined in terms of boundary and region properties of the segments. [71]



**Figure 4.11** - Graph cut segmentation. A) Image with object (O) and background (B); b) Creation of the graph; c) graph partition using edge weights, boundary terms of cost function and positions of seed to obtain optimal minimum cut; d) segmentation result in the image. [80]

There is a variety of methods that are graph cut based as, for example, normalized cuts, random walker, minimum cut, isoperimetric partitioning and minimum spanning tree-based segmentation. Active contours and level sets only use boundary information and they cannot guarantee a globally optimal result. In Graph cut segmentation, the energy function is constructed based on regional and boundary information and it can achieve globally optimal result, being widely used in medical image.

**Markov Random Field (MRF).** MRF is a very used technique for colour segmentation. Components of the colour pixel tuples are considered as independent random variables for further processing. MRF is also combined with edge detection for identifying the edges accurately. [71]

**Concavity points.** Concavity points, polygonal approximation and geometrical model fitting methods require much knowledge about the shape. The concave points split the contours into segments and are mainly used for nuclei segmentation, coupled with watershed. After a contour detection based on morphological filtering and adaptive thresholding, the concave points are detected from the polygonal

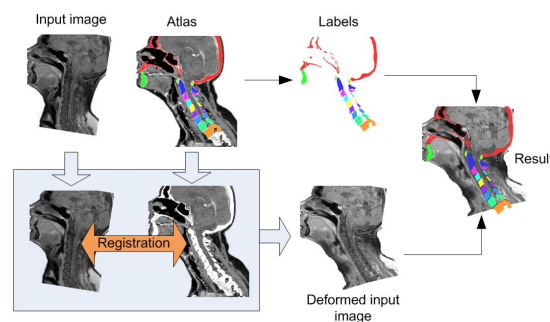
approximation. The advantages of this polygonal approximation are: smoothing, reduced computation time and the critical points are pointed out. [81]

### *Global Priors*

These methods incorporate global shape knowledge about the structure to segment and they enforce the segmentation to be similar to one or a more groups of reference shapes. This way, complex objects can be robustly segmented even in low contrast images. The two mainly types of global prior shape methods are geometric model and voxel atlas based:

**Geometric model based segmentation.** They represent shape as geometric objects like point clouds and polygonal surfaces, simplex meshes, B-spline and level set representations. Multi-resolution Models are based on the analysis at different scales, and a coarse initial segmentation is refined by increasing the resolution. By other side, Deformable Models have the ability to directly generate closed parametric curves or surfaces from images providing robustness to noise and spurious edges.

**Voxel atlas based segmentation.** Two images (a reference image called atlas and the input image to segment) are registered based on the voxel representation of both images. In the first, the structures to segment are already contoured. After registration, both images are in alignment such that the segmented structures in the atlas image can be directly transferred to the input image. This way, all structures in the input image that were labelled in the atlas are segmented, as shown in figure 4.12. [71]



**Figure 4.12-** Voxel Atlas based Segmentation method. [71]

### *Tracing*

In some researches, it may be useful to quantify the numbers, length, and relative size of branching structures in images at multiple scales. It can include vasculature, neurites, and microtubules. The major methods are: skeletonization, obtaining a skeleton structure of the image; vectorizing, a fast method that recursively explores only the regions of interest; and methods that use cylindroids (cylinders with an elliptical cross section as model to represent the properties of the tubular structures).

## **4.6 - Feature Extraction**

In image analysis, computer programs should be able to correctly detect and extract efficient features that belong to the ROIs. The low-level feature detectors are: Edge detectors (Canny, Differential, Sobel, Prewitt and Roberts), Corner detectors (Harris operator, Level curve curvature, SUSAN and FAST), Blobs detectors (Laplacian of Gaussian (LoG) and Difference of Gaussians (DoG)), Ridges detectors and SIFT (Scale-Invariant feature Transform). The usually detected features represent:

- Color Features. A number of color spaces such as RGB, LUV, HSV and HMMD are used with color features such as histograms, color moments (CM), color coherence vectors (CCV) and color correlogram;

- Texture Features. While color is a pixel property, texture is measured from a group of pixels. It is a measure of intensity variation of the surface, which quantifies properties like smoothness and regularity. Requires a processing step to generate the descriptors: Gray-Level Co-occurrence Matrices, loss texture measures, wavelets, and steerable pyramids; and gives information about the spatial arrangement of the color or intensities (mean, contrast, homogeneity, entropy and energy). The most common method is Gabor filter, which is designed to sample the entire frequency domain of an image by characterizing the centre frequency and orientation parameters. The image is filtered with a bank of Gabor filters or Gabor wavelets of different preferred spatial frequencies and orientations. Each wavelet captures energy at a specific frequency and direction that provide a localized frequency as a feature vector. Thus, texture features can be extracted from this group of energy distributions. It's often used as an image descriptor to region based (group pixels based on texture properties) and boundary based (group pixels based on edges between pixels that come from different texture properties) segmentation methods, normally combined with other measures such as color.

- Shape features. Its 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. Figure 4.13 presents an overview of the shape descriptors. [82]

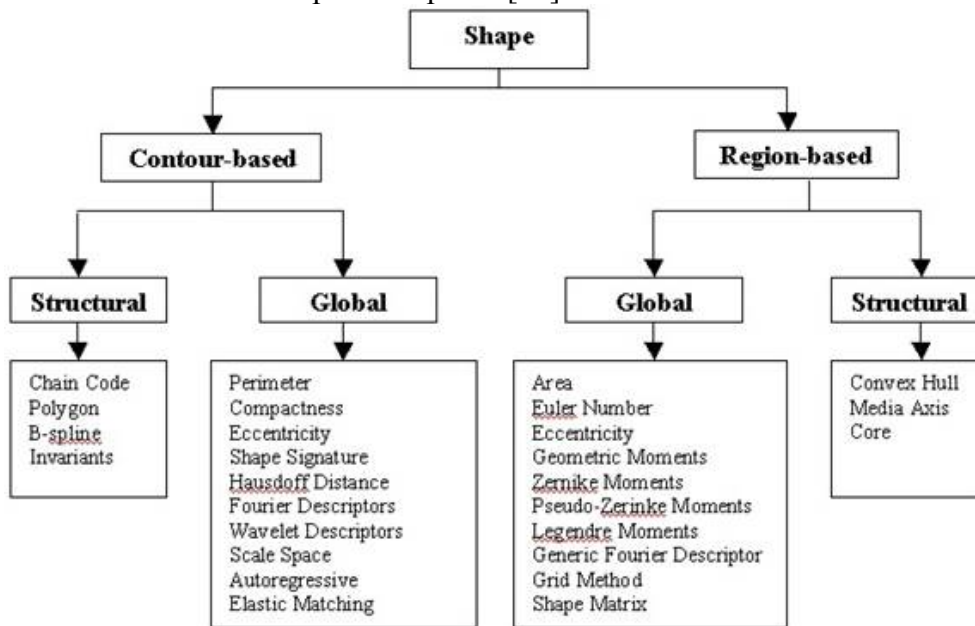


Figure 4.13 – Overview of shape descriptors. [83]

## 4.7 - Clustering and Classification

Artificial Intelligence methods have been widely used in the past decades to implement image segmentation and classification. These algorithms can be divided in supervised and unsupervised methods (Fig.4.14). The first analyse the training data and produce an inferred function to create boundaries between the different objects, while the second doesn't used training data and infer only from image features.

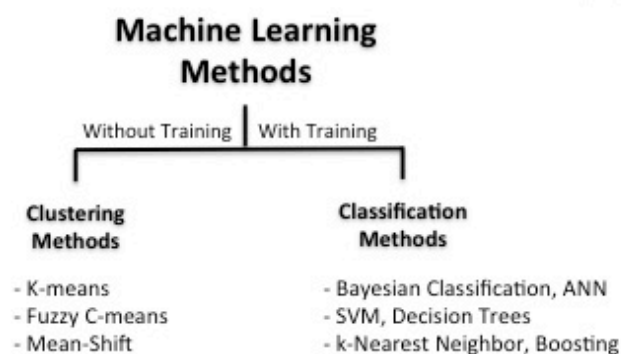
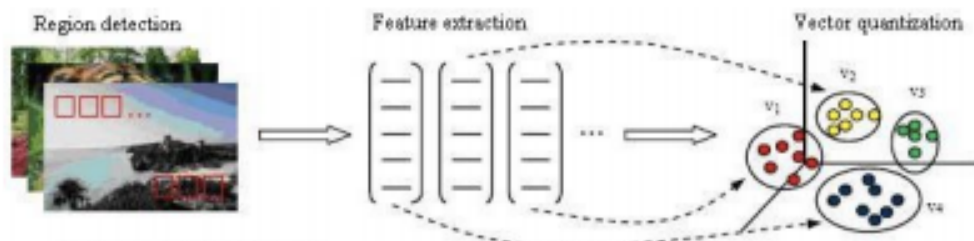


Figure 4.14 – Machine Learning Methods.

The main Classification (supervised) algorithms are: linear classifiers (fisher's linear discriminant, logistic regression, naive bayes classifier, perception), support vector machines (least squares support vector machines), quadratic classifiers, kernel estimation (k-nearest neighbor), boosting (meta-algorithm), decision trees (random forests), neural networks and learning vector quantization. Clustering algorithms can be divided in categorical mixture models, deep learning methods, hierarchical clustering (agglomerative or divisive), k-means clustering, correlation clustering and kernel principal component analysis (kernel PCA).

#### 4.7.1 – Unsupervised Methods

Clustering methods are the most common type of unsupervised algorithms and are based in a representation of the characteristics in the feature space to find natural grouping clusters. The similarity between pixels is measured by the distance to a specific value and pixels with similar distance will be classified as part of the same object. These methods doesn't use training data but it alternates iteratively between segmenting the image and characterizing the properties of each class, resulting in a "self training process" with the available data (Fig. 4.15). It's mainly used when classes are known in advance. The grouping of pixels into clusters is based on the principle of maximizing the intraclass similarity and minimizing the interclass similarity. The most common algorithms for clustering are K-mean (hard segmentation, where each point belongs to a cluster or not) and Fuzzy C-means (soft segmentation, where each point belongs to each cluster in a certain degree). The clustering process can start by considering all data as part of the same cluster and then consecutively dividing until intra-cluster distance is maximized (divisive clustering) or by considering each pixel as a separate cluster and then recursively agglomerating until minimization of the distance between the original clusters. [84]



**Figure 4.15** – After region detection, specific features are extracted and grouped.

**K-means.** It starts with a set of information, and a choice of the number of clusters ( $k$ , number of different regions to segment) that can be determined by counting the modes of the histogram. 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 (Fig. 4.16). 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. [85]

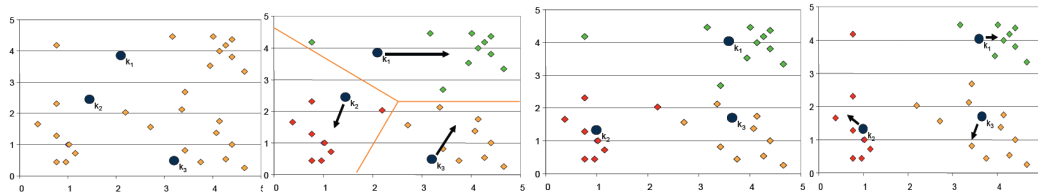


Figure 4.16 - K-means clustering example.

**Fuzzy C-mean.** Fuzzy Clustering divides the input pixels into clusters with some similarity criterion, such as distance, connectivity or intensity. Fuzzy C means (FCM) algorithm is the most accepted method, which preserves much more information than other approaches. FCM is iterative and generates a fuzzy partition matrix and requires cluster centre along with objective function. The values for the cluster centre and objective function are updated in all iterations and stop when the difference between two successive object function values is less than some predefined threshold value. [86]

Other soft computing approach to edge detection is Genetic Algorithm, which consists of three major operations: selection (evaluate each individual and keep the fittest), crossover (recombination of individuals to obtain better ones), and mutation (induces changes in some to maintaining diversification during optimization).

**Mean-shift method.** Finds modes in a set of data samples, manifesting an underlying PDF. This is a non-parametric which needs to estimate to find the modes. For a data point  $x$  in feature space, the density gradient is estimated as being proportional to the mean shift vector. After mean shift filtering each data point in the feature space has been replaced by its corresponding mode. [87]

### 4.7.2 –Supervised Methods

Classifier methods (supervised methods) are pattern recognition techniques that use a set of features with corresponding labels (training) to learn the function and predict the labels for more data. The machine learning framework is obtained with equation 6:

$$y = f(x) \quad (6)$$

In the training process, labelled examples  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  are used to estimate the prediction function  $f$ . This function is then applied to a new test example  $x$  with the image feature and  $y$  is the output.

These methods be considered as linear, with simple applications to distinguish two classes, or non-linear, when the classifier tries to adapt the data to the training as much as possible, to minimize the error. A simple classifier is the nearest-neighbor classifier, in which each pixel is classified in the same class as the training data with the closest value. Figure 4.17 shows the popularity of some classifiers in engineering field. It is possible to conclude that Artificial Neural Networks are gaining much applicability in the last years.

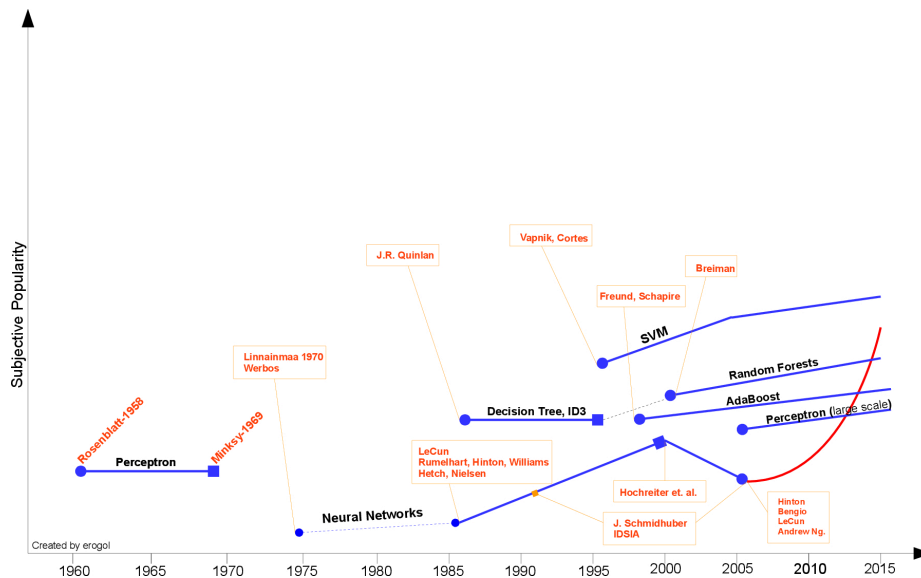
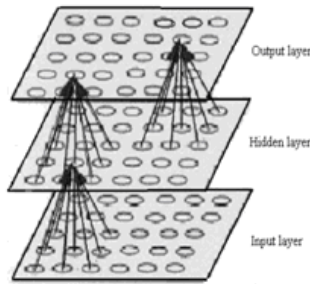


Figure 4.17 – Historical popularity of some machine learning classification algorithms. [88]

**Artificial Neural Networks (ANN).** A neural net is an artificial representation of human brain that tries to simulate its learning process. ANN starts by mapping the

image into a neural network composed of a large number of interconnected processing elements (neurons), that convert the image segmentation problem into an energy minimization problem. The training is done to determinate the connections and weights between nodes. By adapting the weights and bias of the neurons, the algorithm can learn from the training data, improving its performance and adapting to the changes in the environment. Hidden units are nodes that are situated between input and output nodes. They allow the network to learn non-linear functions and to represent combinations of the input features. The number of hidden units should be balanced to avoid overfitting and underfitting. Figure 4.18 shows a simple representation of a neural network. [89]

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 based in the texture, with high accuracy rates, but with extensive supervision, and dependence of the training parameters and absence of noise.



**Figure 4.18** - Neural network approach for Image Segmentation and Classification

**Bayesian Classification.** 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, which is expressed by Equation 7 and 8.

$$P(W_i|x) = \frac{P(W_i)p(x|W_i)}{p(x)} \quad (7)$$

$$p(x) = \sum_{k=1}^c P(W_k)p(x|W_k) \quad (8)$$

In these equations,  $P(W_i|x)$  represents the probability of  $W_i$  occurring given  $x$ .  $P(W_i)$  is the probability alone of  $W_i$  occurring, whereas  $P(x|W_i)$  is the likelihood (training) and

$p(x)$  is the evidence.  $c$  is the number of classes. In other words, the Bayes' formula represents the probability of something happening given the past-known information of its likelihood to occur.

The Bayes' formula is then used to create a Bayesian classifier, described by Equation 9. This "case" style function draws a line according to  $P(W_1|x)$ , which defines if the decision should be classified as part of  $W_1$  or  $W_2$  by the solely probability of occurrence given an  $x$  event.

$$\text{Decision Rule} = \begin{cases} W_1 & \text{if } P(W_1|x) > P(W_2|x) \\ W_2 & \text{else} \end{cases} \quad (9)$$

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. [90] 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.

**Support vector machines.** These classifiers find the best hyperplane that separates all data points of one class from the other, which means the largest margin (with no interior data points) between the two classes (Fig. 4.19). The support vectors are the data points that are closest to the separating hyperplane (on the margin).

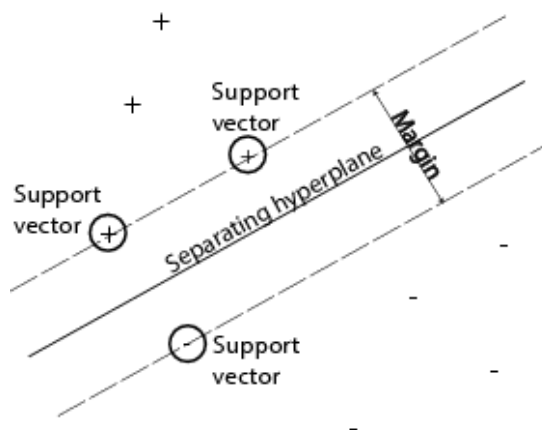


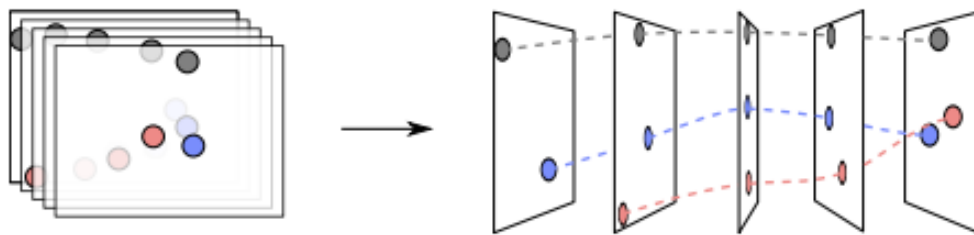
Figure 4.19 - SVM classifier. [91]

There still are supervised algorithms (Ensemble learning algorithms – Bagging, boosting and random forest) that combine multiple learning algorithms together.

Although training is generally associated with supervised methodology, unsupervised networks can also be formulated, e.g., Self-Organizing Maps (SOMs).

## 4.8 - Tracking/Motion Analysis in CAD Systems

Tracking consists in building trajectories of one or several particles as they move along time (Fig. 4.20). To track multiple moving objects from a stack of images, there are two main steps. The first is the detection and segmentation of all particles according to specific techniques, with feature extraction (e.g. location). The second step is the Object Tracking/Motion Correspondence. Given the locations of the particles in successive images, we need to link them along time (frame by frame), taking in consideration that these particles change their location. [6] Due to the large number of biological types with different features and behaviours, designing a universal tracking system is unviable.



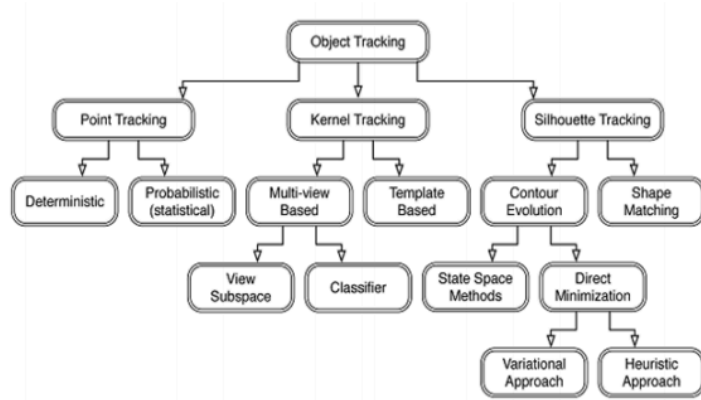
**Figure 4.20** – Multiple moving objects tracking illustration. [92]

### 4.8.1 – Tracking Methods

The Object detection step can be performed with a wide range of image processing methods, such as segmentation methods (normally with mean-shift, graph-cut and active contours), point detectors (with Harris and SIFT detectors), Supervised Classifiers (SVM, Neural Network and Boosting methods) and Background Subtraction. These methods define for each pixel if it belongs to the object or not. Background Subtraction methods, like Hidden Markov models (HMM) and Eigen space decomposition, create a background model and find significant changes in the incoming frames, to consider as a moving object.

In the Object tracking process (Fig. 4.21), the association can be done by considering only points to represent the objects (point tracking), like the vesicle's

centroids, by considering the object shape and appearance (kernel tracking, e.g. mean-shift tracking), and by estimating the object region with shape and contour models (silhouette tracking).



**Figure 4.21** – Object tracking approaches. [93]

The feature selection for tracking methods is an important and subjective step and it's related to the object representation. Many tracking algorithms use color, edges, optical flow and texture features.

The tracking methods can be obtained based in Segmentation (objects have well recognizable boundaries), in Detection (objects have poor contrast but can be detected), in Recognition (objects can be discriminated and recognized based on their features) and in Localization.

There are several data association algorithms that can be divided in Single Data Association approaches (Nearest Neighbor, Track splitting filter and Probabilistic Data Association Filter – PDAF) and Multi-Target Data Association (Nearest Neighbor, Interpretation Tree, Joint Compatibility, Joint Probabilistic Data Association Filter – JPDAF, Multiple Hypothesis Tracking – MHT, and Markov Chain Monte Carlo - MCMC) each having strengths and weaknesses. Table 4.2 presents a comparison between some objects tracking algorithms.

Table 4.2 – Comparison between different moving object tracking techniques. [94]

Methodology	Type of Tracking	Algorithms used	N	Occlusion Handling	Efficiency/ Measurement	Advantages	Limitation
Kalman Filter[2],[9]	Point Tracking	Kalman Filtering algorithm	S	No	Efficiency in terms of total time elapsed(in seconds) for processing certain frames	Used to track points in noisy images	State variables are normally distributed (gaussian)
MHT(Multiple Hypotheses Tracking) [2]	Point Tracking	MHT algorithm	M	Yes	Distance measure is calculated	Able to deal with entries of new object and exit existing object	Computationally exponential both in time and memory
Particle Filter based on Codebook background[2]	Point Tracking	Improved particle filtering algorithm	M	Yes	Get the min. Variance estimate, avg. Processing time/frame is 94ms	Solves the problem of particle degradation of traditional particle filter, the background color interference	---
Dual – Tree Complex Wavelet transform [3]	Kernel Tracking	Dual-Tree CxWT algorithm	S	P	Centroid of the moving object bounding box in each frame is calculated	Good directional selectivity and shape matching	It uses real Filter
Daub Complex Wavelet transform [4]	Kernel Tracking	Daub CxWT algorithm	S	P	Min. Difference of energy of wavelet coefficients between frames	Reduced phase sensitivity and false tracking of objects, helps in preserving the edges	Object shape and size should not change b/w successive frames
Color Histogram[6]	Kernel Tracking	Histogram based algorithm	S	P	Search takes about 500ms and detection rate is 96.5%	Runs very fast, suitable for models having dominant colors	Spatial information of the target is lost, cannot give good performance when an object & its background have similar color
Contour Tracking[2]	Silhouette Tracking	Gradient Descent Algorithm	M	F	Region Statics is calculated using grid points	Object Shape is Implicitly modeled	---
Shape Matching [2]	Silhouette Tracking	Hough Transform	S	P	TSV(Temporal Spatial Velocity) in 4D(x,y,u,v) image per frame is calculated	Less sensitive to appearance variations	It requires Training

N: number of objects being tracked;  
F: full occlusion handling;

S: single object tracking;  
P: partial occlusion handling;

M: multiple object tracking;  
CxWT: Complex Wavelet transform

**NNA Method.** A simple and commonly used approach to link objects is the nearest neighbour association. For an object in the current frame, its distance measure to all other objects in the next frame is computed, and the pair with the shortest distance is linked as the same object. NNA assumes small displacements of the objects and it needs a fixed number of objects and velocity. [7] 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 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. One possible way to obtain it is to use the Hungarian method to link particles between frames based in a combinatorial global optimization (figure 4.22). This algorithm allows to find the minimum weight matching in a bipartite weighted matching matrix, where an infinite weight designates that the considered pair of are not connected. 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.[95]

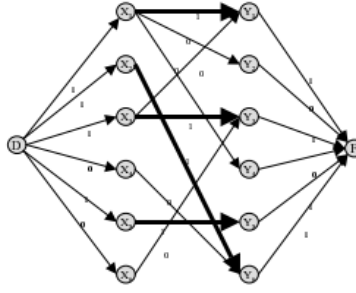


Figure 4.22 – Hungarian algorithm association. [95]

When tracking is performed in an environment where multiple targets can be present, problems related to the validation and association of the measurements arise. In addition to the Nearest Neighbor, which selects the closest measurement, techniques such as Probabilistic Data Association Filter (PDAF) are available. The underlying assumption of the PDAF is that for any given target only one measurement is valid, and the other measurements are modelled as random interference (uniformly distributed random variables). The Joint Data Association Filter (JPDAF), on the other hand, calculates the measurement-to-target association probabilities jointly across all the targets.

A different strategy is represented by the Multiple Hypothesis Filter (MHF), which evaluates the probability of a given target to give rise to a certain measurement sequence.

**Kalman Filter.** In dynamic systems, the objective of tracking is to estimate the state, given all the measurements  $z_{1:k}$  up to that moment, or equivalently to construct the pdf  $p(x_k | z_{1:k})$ . The theoretically optimal solution is provided by the recursive Bayesian filter, which solves the problem in two steps, prediction and update. The prediction step uses the dynamic equation and the computed pdf of the state at time  $t = k - 1$ ,  $p(x_{k-1} | z_{1:k-1})$ , to derive the prior pdf of the current state,  $p(x_k | z_{1:k-1})$ . Then, the update step employs the likelihood function  $p(z_k | x_k)$  of the current measurement to compute the posterior pdf  $p(x_k | z_{1:k})$ .

When the noise sequences are Gaussian, the optimal solution is provided by the linear Kalman filter, which yields the posterior to be also Gaussian. In nonlinear systems, Extended Kalman Filter (EKF) are used, and when the state space is discrete and consists of a finite number of states, Hidden Markov Models (HMM) filters can be applied for tracking.

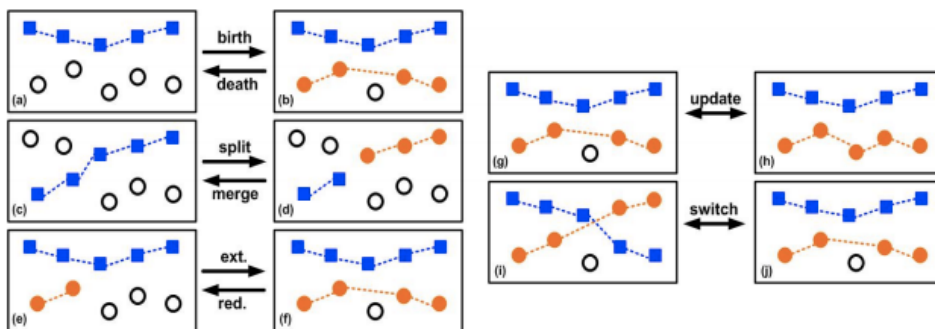
**Particle Filters.** The most general class of filters is represented by particle filters, also

called bootstrap filters, which are based in the Monte Carlo integration methods. The current density of the state is represented by a set of random samples with associated weights and the new density is computed based on those samples and weights. The Unscented Kalman Filter (UKF) can be employed to generate proposal distributions for particle filters, in which case the filter is called Unscented Particle Filter (UPF). [96]

#### 4.8.2 – Vesicle Tracking

The acquisition of vesicles in images with fluorescent markers in neuronal cultures, using optical imaging, normally possesses some common characteristics: 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 leads to false positives (background noise classified as vesicle) or false negatives (missed detection of real vesicles), compromising the quantification. [7, 97]

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's 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. [7, 98] Due to this nonlinear movement and events, non-linear approaches are the best solution to track these biological targets. Figure 4.23 presents an example of a tracking in a dynamic object environment.



**Figure 4.23** – Illustration of object association over time with some dynamic problems. [99]

### 4.8.3 – Manual 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 user-biased 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. [100, 101]

### 4.8.4 – Particle Tracking Tools

In table 4.3 some Particle Tracking Tools available in the market are presented. For each it's represented the available form (freeware, paid or by requesting the developers), the platform in which the software runs (Java, ImageJ, IDL, Matlab, Linux, Mac and Microsoft Windows), the maximum spatial dimensionality per frame it can handles (2D or 3D) and the level of automation (automatic, manual or semi-automatic).

### 4.8.5 – 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 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. [102, 103]

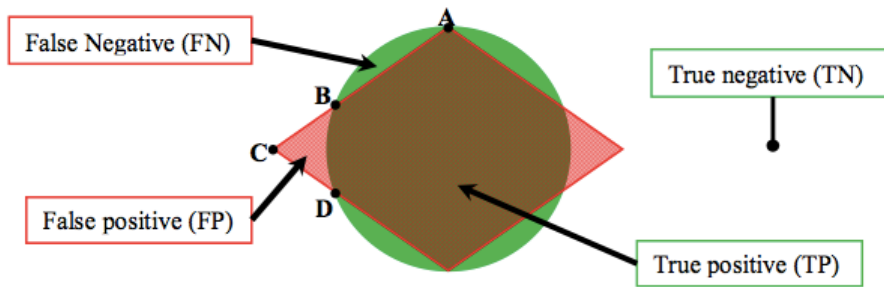
Table 4.3 — List of some available particle tracking tools. Adapted from [101].

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

## 4.9 - 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. According to the classification of the algorithm, four types of labels can be obtained to the candidates (Fig. 4.24):

- 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.



**Figure 4.24** – Detected objects and have 4 types of labels (TP, TN, FP and FN). [104]

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. [71,72]

**Accuracy.** The accuracy (expressed in Equation 10) is a global performance measurement of the algorithm that shows the percentage of correct classifications.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

**Sensitivity.** The sensitivity or recall (described in Equation 11) indicates the true positive fraction, i.e., from all positives, which percentage the algorithm considers.

$$\text{Sensitivity} = \text{Recall} = \frac{TP}{TP+FN} \quad (11)$$

**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 is represented in Equation 12.

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (12)$$

**Precision.** The precision (represented in Equation 13) indicates which fraction of the detections is relevant, i.e., how many of the positive detected are really lesions.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

**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  $\beta$  values (that normally varies from 0.5 to 2). When  $\beta = 0.5$ , precision counts double of the recall, and the inverse situation for  $\beta = 2$ . With a F-value higher than  $\beta = 0.5$ , 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. The F-measure is expressed in Equation 14.

$$F_{\beta} = (1 + \beta^2) \frac{P \cdot R}{\beta^2 P + R} \quad (14)$$

**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 4.25). 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.

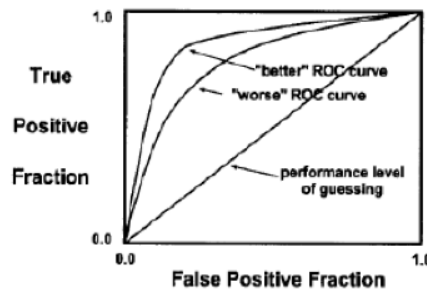


Figure 4.25 – ROC Curve representation. [105]

Between two ROC curves, we choose the one that presents the higher Area Under the Curve (AUC), possessing higher capacity to take correct decisions. However, it is needed to guarantee that a higher area does not result in less true positives. [105]

To evaluate the tracking method, the number of correct complete paths measured is related with the total number of complete paths in a simple metric represented in equation 15.

$$Tracking = \frac{\#correct\ complete\ paths}{\#complete\ paths} \quad (15)$$

**Part II**  
***NeuronDyn*: Vesicle Movement Characterization**  
**Software**



# Chapter 5

## *NeuronDynamics*

As explained in the Introduction, this project started in a CAD course from the Master Integrated in Bioengineering. In a six-week project with the collaboration of three colleagues, a semi-automatic algorithm for vesicle movement characterization was elaborated in Matlab. This first version of the *NeuronDyn* will be explained in this chapter coupled with the discussion about the results and the improvements to *NeuronDyn* approach.

### **5.1 - *NeuronDynamics*: A Method for Neurotransmitter Vesicle Movement Characterization in Neurons**

*NeuronDynamics* is a novel method for the tracking of fluorescent-marked vesicles along an entire image sequence. It's based on a recursive Bayesian estimation to detect vesicles and a nearest neighbor association to link them frame-to-frame. It also relies on a previous user-controlled training stage that adapts the analysis to the dataset.

#### **5.1.1 – 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  $399 \times 201$  pixels<sup>2</sup> and a video frame rate of one frame per second. The number of vesicles (true positives) ranged between 25 and 29.

### 5.1.2 – 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. [106] 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. [90]

### 5.1.3 – 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 Equations 16 and 17:

$$I_p(i, j) = \begin{cases} I_{\text{LoG}}(i, j) & \text{if } I_{\text{LoG}}(i, j) > T \cdot \max\{I_{\text{LoG}}(i, j)\} \\ 0 & \text{else} \end{cases} \quad (16)$$

$$I_{\text{out}}(i, j) = I_p(i, j) - I_{\text{in}}(i, j) \quad (17)$$

Where  $T$  is a parameter to determine the threshold level,  $I_{\text{LoG}}(i, j)$  is the output image from the enhancement step,  $I_p(i, j)$  is the image after the threshold operation and  $I_{\text{in}}(i, j)$  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.

### 5.1.4 – 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 equation 18:

$$S_{v,k \rightarrow k+1} = \frac{D_{v,k \rightarrow k+1}}{\text{Video Rate}} \quad (18)$$

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 5.1 is a pipeline of the algorithm.

### 5.1.5 – Algorithm Evaluation

The algorithm was compared to FluoTracker [7], a automatic method for vesicle movement characterization, based on accuracy, sensitivity, specificity, precision, F-value and computational time. This comparison was performed by using a ground-truth established by the authors following technical instructions from researchers with field experience. In each frame (without pre-processing) of one of the two available films, the particles that were confirmed as vesicles were marked. The second film was used to calculate all the possible candidates, the number of vesicles between detections and the algorithm's performance.

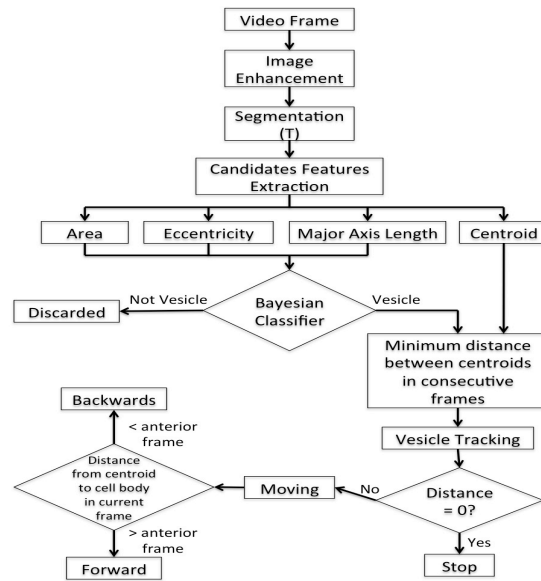


Figure 5.1 – NeuronDynamics' pipeline.

## 5.2 - Discussion and Improvements

*NeuronDyn* will be structured based in *NeuronDynamics* fundamentals but with some improvements that are expected to complete the study and to obtain better results. Here will be presented the main problems that exist in this approach, and the solutions that are going to be implemented.

**Vesicle Detection and Segmentation.** Vesicle Detection is the core of the tracking process, that has some issues: 1) The recordings are often noisy, 2) True vesicles can be left behind due to size specifications, 3) Information could be lost from one frame to another, 4) User bias during the training phase, 5) Incorrect selection of T. To avoid some of these problems, its necessary to continue with the training stage, performed by an expert supervision that will help the Classifier's performance.

Other important aspect in this topic is to automate the detection of the optimal T for each dataset. An automated detection of the optimal T parameter will force the algorithm to always have its best true positive fraction for each analysis, thus reducing the risk of poor results. This parameter should be obtained taking in consideration that it is more desirable to have a smaller number detected vesicles that are true positive, rather than detecting a lot of vesicles without being able to perform a clean tracking.

Other important aspects of this topic that are:

- Before the training stage, the user should have an option to see the entire film and choose in which frame he wants to do the training stage. Taking in consideration that

normally the users mark 10 vesicles and that in some frames the number of perceptible vesicles is lower than that, this is an important improvement. After this, the chosen frame should be subject to a visual enhancement operation to increase the contrast between the vesicle (fluorescence region) and the background;

-Taking in consideration that some confocal images can have more than one neuron, the study of vesicle movement characteristics should have an option to consider each individual process (axons and dendrites) and, of course, for the entire image too.

-For a set of films obtained with the same parameters of acquisition (resolution, fluorescence, etc.) that are going to be analysed, only one global training stage should be enough.

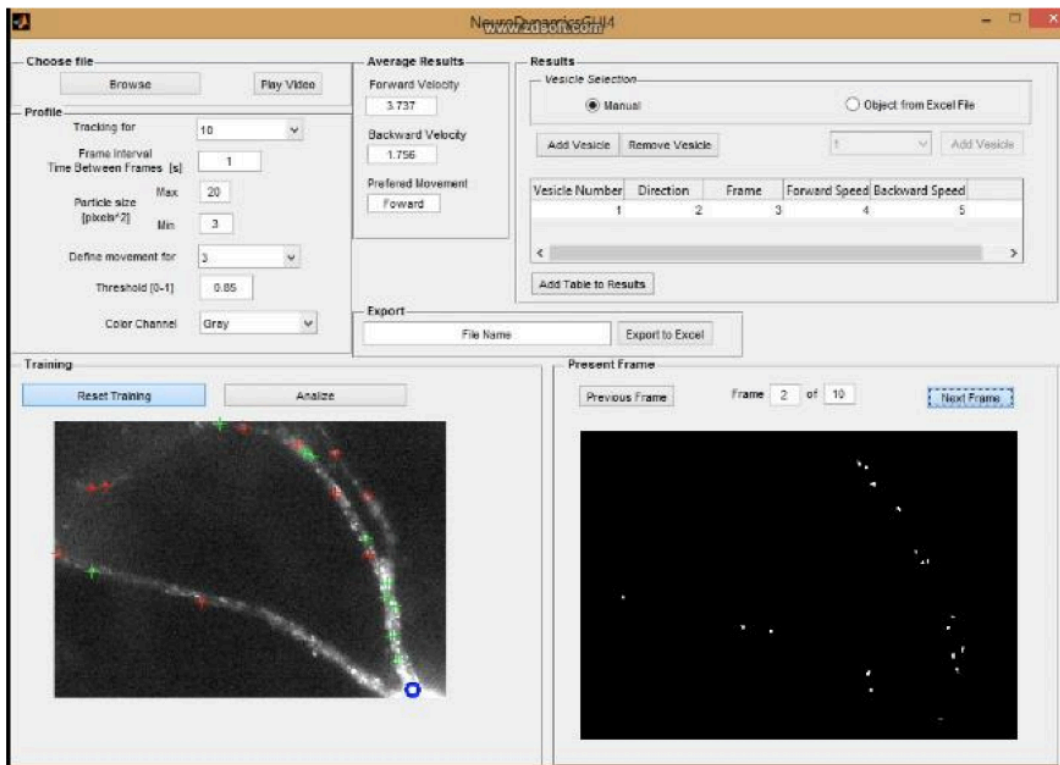
**Tracking.** The principle behind tracking in *NeuronDynamics* method is NNA. It consists in assigning each vesicle from the present frame to a coordinate based on the nearest vesicle in next frame. But due to the dynamic nature of neuronal vesicles (merge/split, birth/death, clustering), the next position can be misdetected. The tracking will occur and the vesicle will be associated to the nearest vesicle position, whether it is its true correspondent or not. In *NeuronDyn* approach, a more complete tracking method to consider these phenomena should be added.

**Evaluation.** *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 figure 5.2. 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.

Algorithm	Accuracy	Sensitivity/Recall	Specificity	Precision	$F_1$	$F_{0.5}$	$F_2$	Computation Time (s)
<i>NeuronDynamics</i>	0.853	0.849	0.956	0.978	0.818	5	1.25	0.183
<i>FluoTracker</i>	0.414	0.434	0.403	0.285	0.344	5	1.25	125.2

**Figure 5.2** – Algorithm Performance Comparison for *NeuronDynamics* and *FluoTracker* using default settings

Another improvement introduced regards the interface with the user, allowing a better interaction. The *NeuronDynamics* interface made in GUIDE from Matlab can be consulted in figure 5.3, while the excel sheet of the results it's in figure 5.4.

Figure 5.3 – *NeuronDynamics* Interface in Matlab.

Microsoft Excel - Isbi\_2014 [Modo de Compatibilidade]

NeuronDynamics 1.0  
 10-Oct-2013 18:40:52  
 #Frames: 10 | #Vesicles in Training: 5 | Area: [3,20]  
 #Detected Vesicles: 5  
 Preferred movement: FORWARD  
 Forward Velocity: 3.737 pixels/s  
 Backwards Velocity: 1.756 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	0	0	0	0	0	0	0	0	0	0
Obj. 2	0	0,1	0,223607	0,761577	0	0,223607	0	0	0,412311	0,2	0,19211
Obj. 3	0	0,1	0	0	0,1	0	0	0,412311	1,104536	1	0,271685
Obj. 4	0	0,223607	1,140175	0,538516	0	1,923538	0,282843	0,282843	0,1	0	0,449152
Obj. 5	0	0	0	0,223607	0	0	0	0,424264	0	0,565685	0,121356

**BACKWARDS 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	0	0	0	0	0	0	0	0	0	0
Obj. 2	0	0	0	0	0,72111	0	0,608276	0,282843	0	0	0,161223
Obj. 3	0	0	0,360555	0,640312	0	2,002498	0	0	0	0	0,300337
Obj. 4	0	0	0	0	0	0	0	0	0	2	0,2
Obj. 5	0	0,1	0,2	0	1,421267	0	0,1	0	0,141421	0	0,196209

**PREFERRED MOVEMENTS BY VESICLE**  
 Movemen F: forward, B: backwards, E: equal, S: stopped

Obj. 1	S
Obj. 2	F
Obj. 3	F
Obj. 4	F
Obj. 5	B

Figure 5.4 – Example of *NeuronDynamics*' results..

## Chapter 6

### ***NeuronDyn* – Vesicle Movement Characterization Software**

As explained in the previous chapters, fast vesicle tracking is a need for investigators in the molecular biology field, and its automation and simplification leads to an increasing interest in the computer vision investigation.

In this chapter the NeuronDyn solution will be explained taking in consideration the problem addressed, the challenges and difficulties, the direction and all its modules/methods. This will give the reader important knowledge about the software obtained for further results, discussion and application.

#### **6.1 - Introduction**

The goal of this research is to perform vesicle segmentation and tracking in 2D confocal videos. The computational algorithm proposed has a supervised Artificial Neural Network (ANN) training approach for classification and a global nearest neighbor tracking method based in Hungarian algorithm to associate the vesicles over sequences of confocal images. This research has been performed in cooperation between the *Brain Lab* investigation group of *INESC-TEC* and *IBMC* (through Dra Paula Sampaio), and it was performed in the two institutions. The software was programmed on MAC-OSX using Matlab R2013a with Image Processing and Neural Network Toolboxes.

## 6.2 - Problem, Challenges and Difficulties

Confocal datasets include multiple long image sequences with a large number of vesicles that show many dynamic properties over time, making manual annotation hard to obtain. This problem gives a challenging task to obtain softwares that can adapt and extend the available and trained images to the rest of the sequences, improving the researchers work.

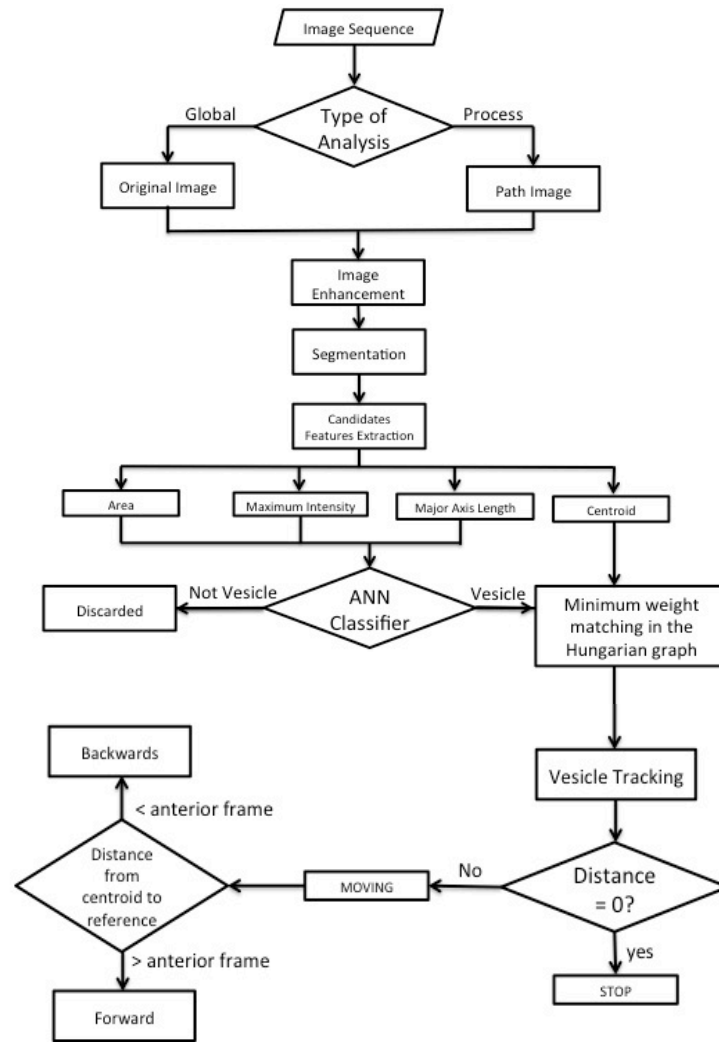
Although staining techniques are used to increase the contrast between vesicles and background, they can cause undesirable stained regions within the axon path hindering its analysis. Also, to keep the neurons alive and healthy, the light exposure must be carefully controlled to minimize phototoxicity.

Normally, confocal datasets present images corrupted with a lot of noise (resulting from the acquisition process), low contrast (from limited light exposure) and other problems, which makes the use of image processing algorithms indispensable. But the key challenge of problem addressed is in the tracking process. Vesicles are very small structures that can be presented as small points in confocal images, which makes it hard to apply shape features in the algorithm. Their dynamic phenomena are other crucial problems, making the temporal correspondence a very difficult task to obtain with high precision.

## 6.3 - NeuronDyn Modules

In order to respond to these problems and to address the investigators needs, a multi-stage approach was produced, divided in Training, Vesicle Segmentation and Classification, Vesicle Tracking and Quantification.

The NeuronDyn approach is schematized in the following block diagram (figure 6.1):



**Figure 6.1** - *NeuronDyn* pipeline.

The algorithm starts with the upload of the video and selection of process or global analysis. Then, the frame to perform the training stage where vesicles and non-vesicles objects are marked. Based on the classification by the user, *NeuronDyn* performs the segmentation and extracts the candidate's characteristics. The classifier discards the candidates that are not vesicles and the algorithm performs the tracking to obtain some measures. The *NeuronDyn* UI is presented in figure 6.2. Each block of *NeuronDyn* is now presented with all its features. Their presentation will be shown considering the interface steps, to follow the path taken by researchers in the interaction with the algorithm.

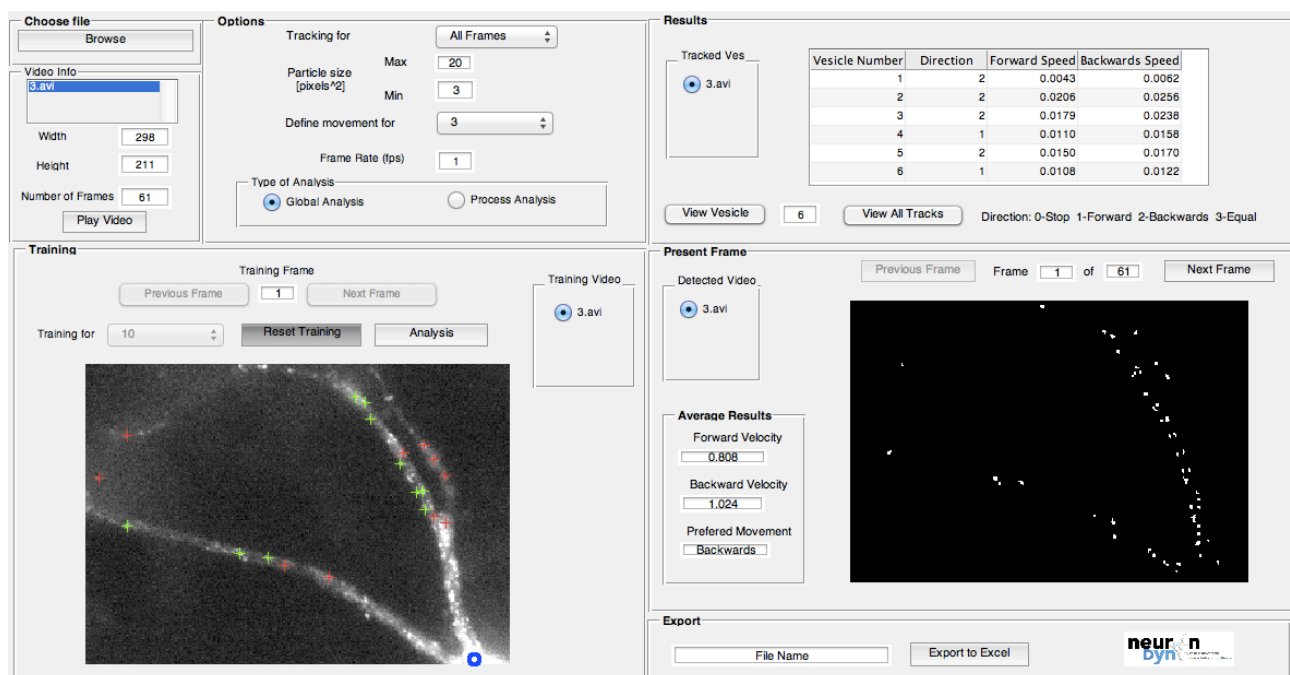
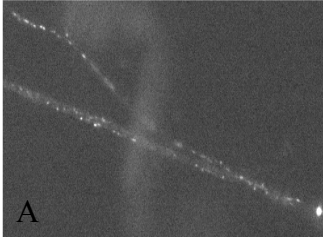
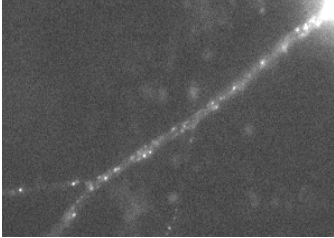
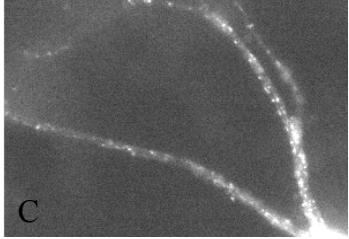


Figure 6.2 - NeuronDyn UI.

### 6.3.1 – Dataset Videos

The videos used for testing *NeuronDyn*'s modules are composed of 3 videos provided from the Department of Functional Genomics, Vrije University, Amsterdam, with neurons conducting marked vesicles (NPY-EGFP) (Datasets 1,2 and 3) all obtained by confocal microscopy. If nothing is said in contrary, the images presented from now on refer to the first frame of the corresponding dataset. The three videos and its characteristics can be observed in table 6.1.

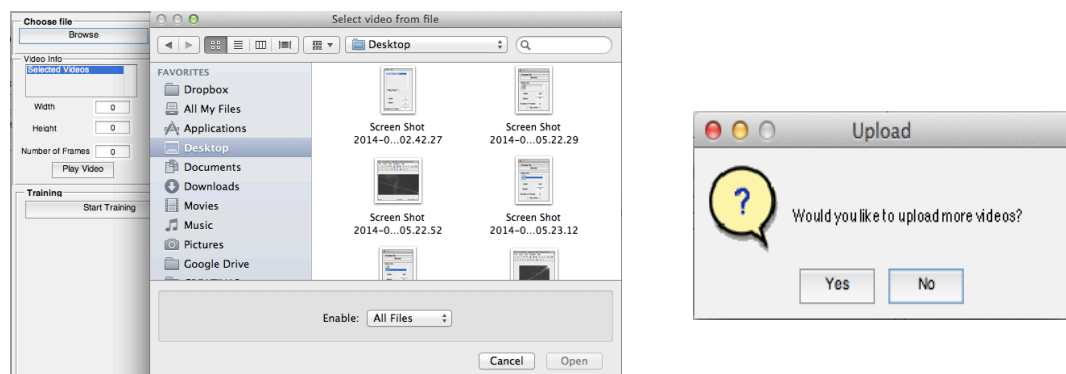
Table 6.1 —Datasets used for *NeuronDyn*.

		
<b>A) Dataset 1</b>	<b>B) Dataset 2</b>	<b>C) Dataset 3</b>
Frame size 399x201;	Frame size 298x187;	Frame size 298x211;
Frame Rate 1 fps;	Frame Rate 1 fps;	Frame Rate 1 fps;
Number of Frames = 61	Number of Frames = 61	Number of Frames = 61

### 6.3.2 – Input Videos to *NeuronDyn*

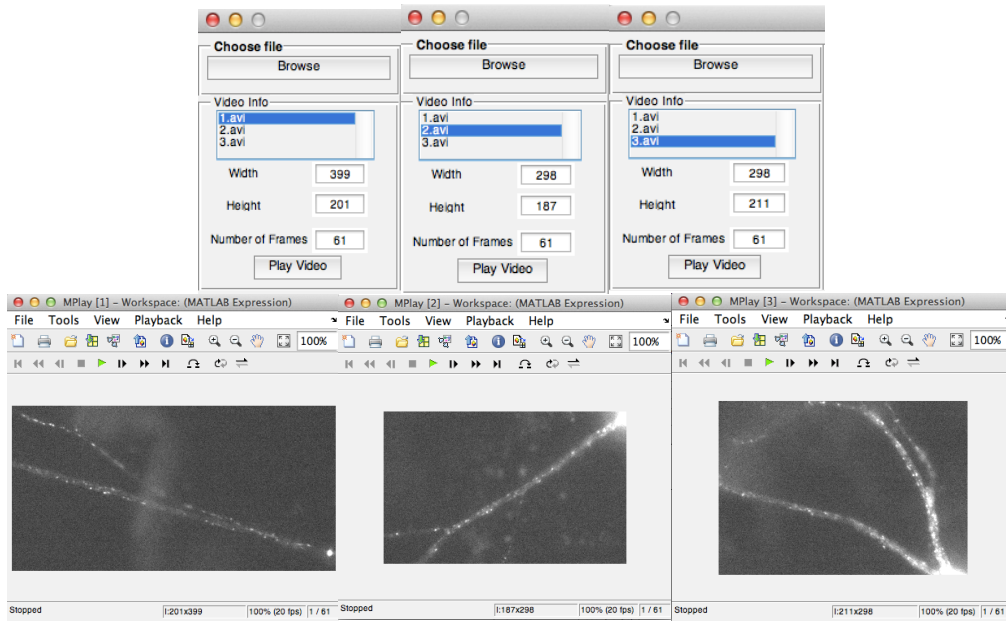
The first phase of the *NeuronDyn* algorithm is to upload one or more videos (up to three), obtained with the same characteristics. This allows the user to run the algorithm and to do the training stage only one time, saving time. To be possible, all the videos should be acquired from the same samples so that the classifier performs well with the marked objects.

The initial block to upload the videos is constituted by a browse button and video information. After the selection of the first video, a message window will appear, to ask the user if he wants to upload more videos (figure 6.3). Then when each one is loaded, the name will appear in the selected videos list, where it's possible to see its width, height and number of frames by clicking in the name. Also, it's possible to play each video in a different window or even all of them at the same time (figure 6.4).



**Figure 6.3** - Upload videos interface in *NeuronDyn*. a) Selecting video from the file; b) message after each uploaded

video.



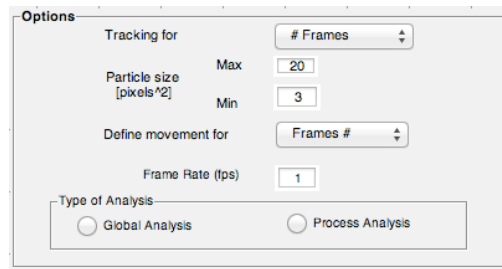
**Figure 6.4** - Characteristics and display of the selected videos.

Frequently, investigators have to analyse different videos from the same cell cultures, acquired with the same characteristics (like the examples shown above). Taking this in consideration *NeuronDyn* allows uploading up to three videos and performing only one training stage. The ANN classifier used has the ability of learning from the training data, improving its performance and adapting to changes in the environment. This new approach simplifies the work of the researcher without compromising the results obtained once the objects have similar characteristics.

For each video a set of characteristics are extracted (like height, width and number of frames).

### 6.3.3 – Parameters Selection

In this step the user can select some parameters that will affect the tracking process (Fig. 6.5). The number of frames to track (all or just specific number), the area of the particles to consider in the process (minimum and maximum value in pixels<sup>2</sup>), the minimum number of consecutive frames to consider as not-moving vesicle (typically 3 frames), the frame rate, and the type of analysis: Global, for the entire image, and Process analysis, for a small portion of an axon (between two selected points).



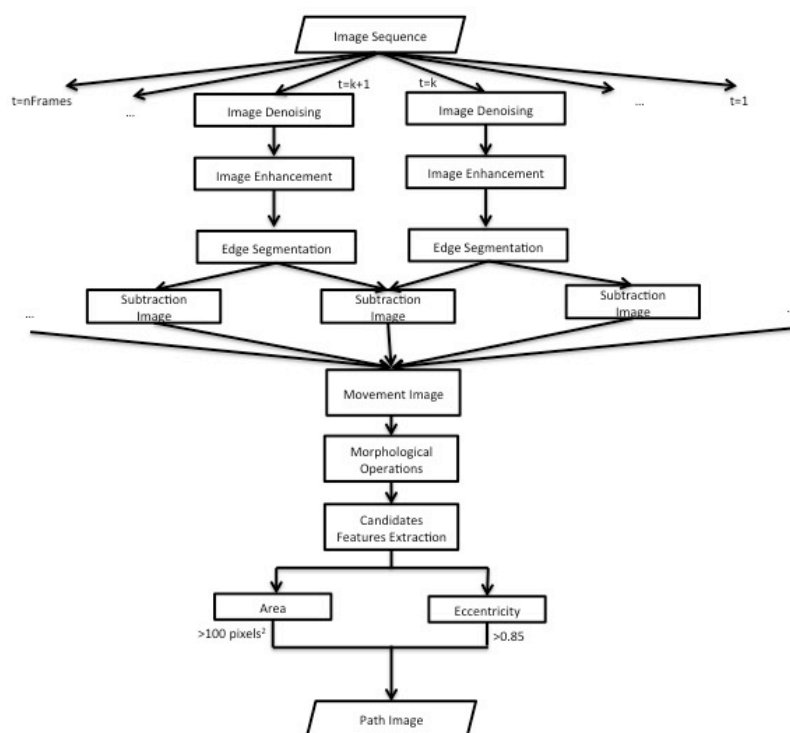
**Figure 6.5** - Tracking parameters of *NeuronDyn*.

### *Process Analysis*

Sometimes researchers only want to analyse and take conclusions about a single neuron process. To obtain it, an option was created in *NeuronDyn*, which also as benefits in the processing by decreasing the computational cost of the algorithm as only a small part of the image is analysed. The main steps of the Process Analysis are Path Detection and Process Selection:

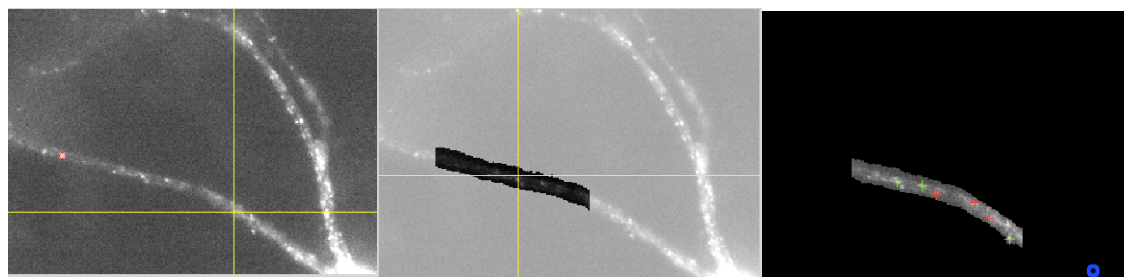
**Path Detection.** To analyse the video by processes, first we need to know in which part does the vesicles move to create the path image of the neurons. To obtain it, the difference between two consecutive frames for the entire image sequence is considered. But first each image gets denoised and enhanced to prevent the presence of unwanted elements in the final image, like noise and stains. The denoising filter selected was Wiener filter, which allowed to remove the 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 resulting are overlapping 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  $\text{area} < 100 \text{ pixels}^2$  and with  $\text{eccentricity} < 0.85$  are eliminated. As explained before, axons are long tube structures with the same diameter as it extend, presenting an eccentricity  $e \approx 1$ , which makes it unchanged by this conditions. Figure 6.6 shows the pipeline of the Path Detection algorithm implemented for *NeuronDyn*.



**Figure 6.6** - Path Detection pipeline.

**Process selection.** To select the process to analyse, the user is asked to select a starting and end point in the original image. Figure 6.7 shows an example of that selection, and the following region in the path image and the obtained image for training. Between the two selected points a region-growing (or in this case “path-growing”) function was create by the author 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 correct points.



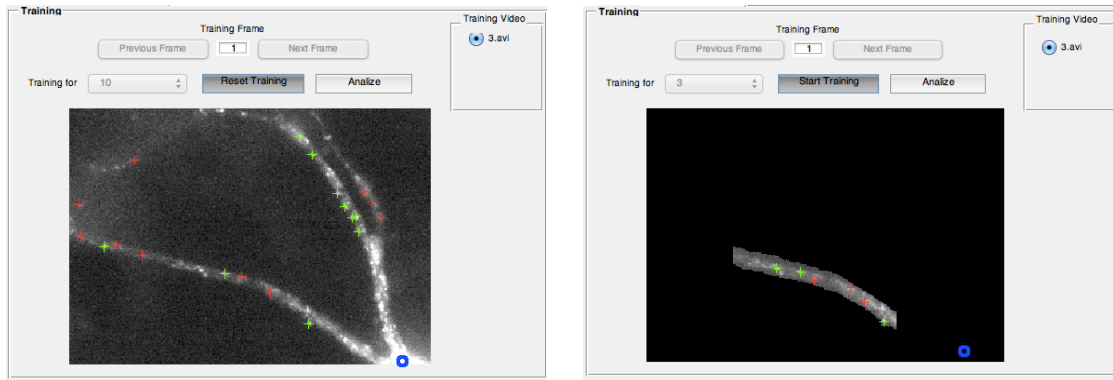
**Figure 6.7** - Process selection and vesicle training detection.

### *Global Analysis*

In the global analysis, the training algorithm is performed in the whole image, subjected to a visual enhancement operation (linear expansion of the histogram).

### 6.3.4 – Algorithm Training

In the training block, the user can visualize each frame of the selected video to choose in which he wants to do the training stage. Taking in consideration that normally the user marks 10 vesicles and that in some frames the number of perceptible vesicles is lower than that, this represents an important feature. Then, the chosen frame should be subject to a visual enhancement operation to increase the contrast between the vesicle (fluorescence region) and the background. 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 if it is present). The expressed method can be observed in figure 6.8.

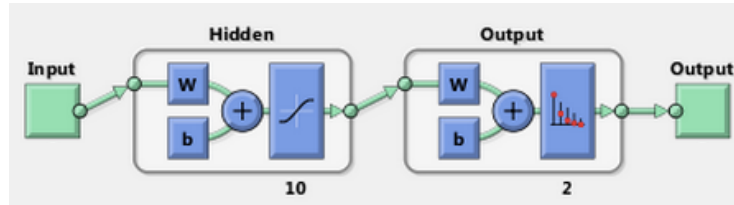


**Figure 6.8** – *NeuronDyn* Training section: a) Global analysis and b) Process analysis.

Each user-marked point is then used as a seed point to a region growing segmentation algorithm that allows the extract the centroid coordinates, major axis length and maximum intensity value. [82] The training method uses an automatic feed-forward ANN classifier (figure 6.9) with one hidden layer of ten units (neurons) and a output layer with two neurons, as there are two target categories associated with each input vector (vesicle or non-vesicle). [70] The classification is achieved based on the major axis length and the maximum intensity values. This approach allows the algorithm to perform a non-linear classification and to build a network without any parameter setting by the user. When an input vector of a vesicle is applied to the network, the corresponding neuron should produce a 1, and the other neuron output a 0. Unlike the first version of the work (*NeuronDynamics*), which uses a Linear Bayes Normal Classification, dividing the two classes with a straight line, *NeuronDyn* classifier can separate unpredictable classes' shapes. Also, as more than one video can be applied, this method will allow to learn from the training data and adapt to the changes verified. This automatically trained feed-forward neural network classifier uses the Levenberg-Marquardt algorithm for training, modifying the values of the

weights in a group way, to minimize the network error.

As negative point, the computational cost applied will increase, but as the number of neurons selected is small that cost will not be very high. This number was selected taking in consideration that large number of neurons is associated with very complex problems that require much more learning.



**Figure 6.9** – Neural Network structure used for classification, where the number of neurons in the hidden layer is 10 and the number of outputs is 2.

### 6.3.5 – Vesicle Segmentation and Classification

Similarly to *NeuronDynamics* approach each frame of the video, disregarding all objects next to the borders and with areas out of a user-predefined range, is 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 Equation 16 and 17 (see section 5.1.3). But unlike that approach where  $T$  had a fixed value (only optimal for the tested datasets), in *NeuronDyn* the threshold is obtained by considering the middle value between the mean intensity of the marked true vesicles and the same from the ambiguous candidates. If the ambiguous candidates are objects with intensities near the true vesicles, the middle value between the two classes will produce a good segmentation result.

For each segmented object the major axis length and the maximum intensity value parameters were obtained, as well as the coordinates for its centroid. The objects are then classified using the fore-mentioned classifier.

### 6.3.6 – Vesicle Tracking

Investigators prefer to detect only few real vesicles instead of a large number with some non-vesicles. Taking this in consideration, the low quality of the images and the differences of fluorescence on the same object between frames, the tracking module was designed. As only “strong” vesicles are detected, one common problem verified is the detection failure in some frames. To overcome this appearance of gaps in the tracks,

an optimal generative association for multi-target tracking based on the Hungarian method was performed. The data association in *NeuronDynamics* was performed using the simple Nearest Neighbor association, which simply links the closest detection in the next frame, mismatching the objects that came from gaps.

This new Global Nearest Neighbour tracking algorithm performed in *NeuronDyn* consists in re-building the trajectories of the particles as they move along time, using the Hungarian method to link the particles in a combinatorial global optimization. The algorithm allows to formulate a bipartite weighted matching matrix, finding the minimum weight matching.

In the obtained tracks, from each point to the next one, the vesicle speed (in pixels per second) is estimated using equation 18 (see section 5.1.4). The vesicle is then 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 (or the reference point, in the absence of cell body in the image) than the last one, the vesicle is labelled as moving forward. If opposite, backwards, but can also be classified as 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. These results, 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.

The outputs are then exported to an Excel file, according to the selected parameters by the user. In default case, the excel will be saved with the name of the program (*NeuronDyn*), and the date. The number of Excel sheets is proportional to the number of analysed videos.

The function used to transfer the information from Matlab to Excel in *NeuronDyn*, was performed using Apache POI java library, allowing the visualization of the document in the different platforms supporting java: Mac, Linux and Windows.

### 6.3.7 – Algorithm Evaluation

The segmentation of the algorithm was evaluated based on accuracy, sensitivity, specificity, precision, F-value and computational time and compared with *NeuronDynamics* and *FluoTracker* results. It was performed using a ground-truth established by the author following technical instructions from researchers with field experience in the last year. Two films of 10 frames each, 399x201 pixels<sup>2</sup> and 1 fps, from Dataset 3 were used to perform the evaluation (Training Set + Testing Set). The number of vesicles (true positives) ranged between 25 and 29. In each frame (without pre-processing) of one of the two selected films, the particles that were confirmed as vesicles were marked. The second film was used to calculate all the possible

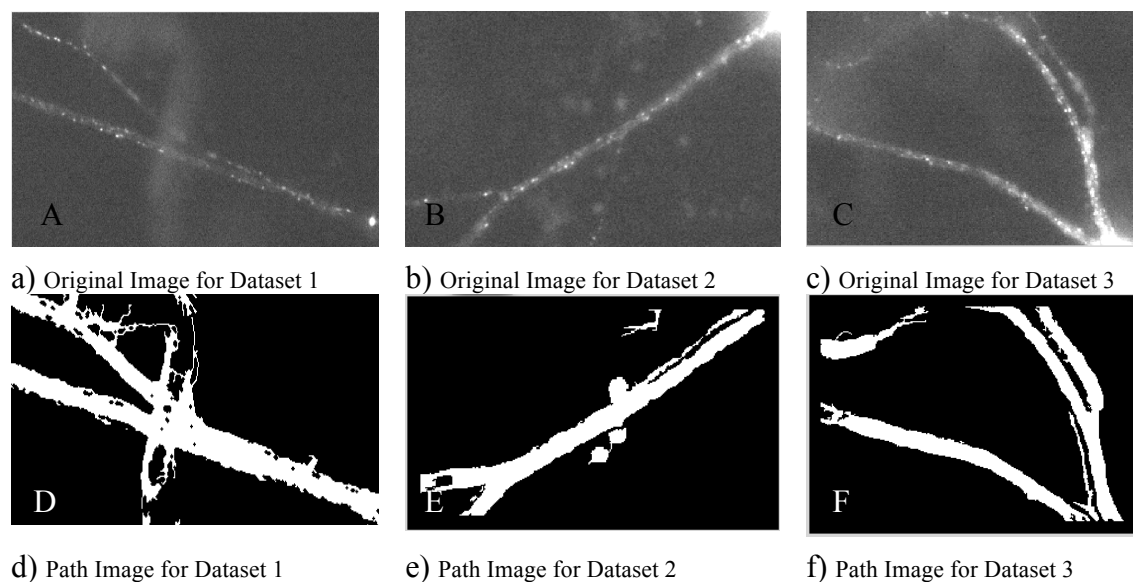
candidates, the number of vesicles between detections and the algorithm's performance.

## 6.4 - Results and Discussion

### *Process Analysis*

In the process analysis module, different types of accuracy can be obtained in the path Image, according to the artifacts presented in the images, the frame rate and the number of frames.

Figure 6.10 shows the final path image obtained for three input videos (Datasets 1,2 and 3). In Dataset 1, d), the produced image presents a bad result, due to the stain from the original image sequence. In the second case, Dataset 2, e), the original stains are almost well suppressed, producing a good image. In the last example, Dataset 3, f), the visual result is good.



**Figure 6.10** – Original and Path images obtained for datasets 1,2 and 3..

If the number of frames considered is very low (for example 5 frames) the same process will produce much poor results, as it can be observed in figure 6.11, which invalidates this approach.



**Figure 6.11** – Path detection for the first Data set using only 5 frames.

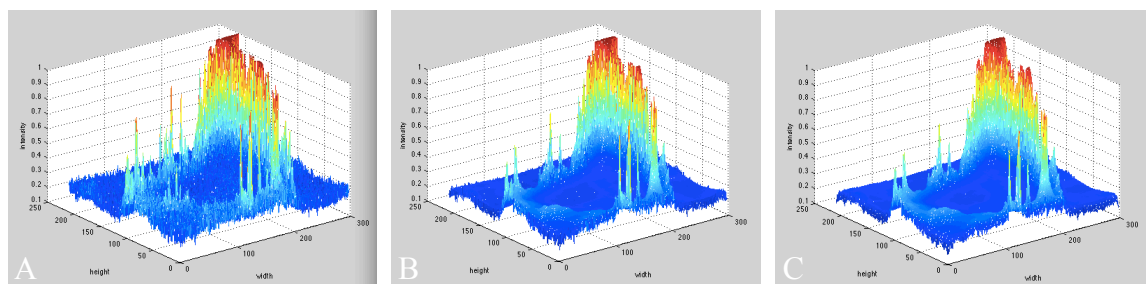
As it is possible to visualize, there are a lot of problems that can hinder this method:

- clusters/cell body or objects that do not move;
- Low frame rate videos, which cause high displacements, resulting in bad path detection;
- Videos with a low number of frames;
- Stains and background noise.

As dataset 3 produced the best results in the previous sections, and due to its facility of processing (low resolution and number of frames), from now on, a complete description is shown for this dataset.

### *Denoising and Enhancement*

To reduce the background noise a sequence of Weiner and Gaussian filters is used, producing visual improvements that help the image analysis (Figure 6.12).



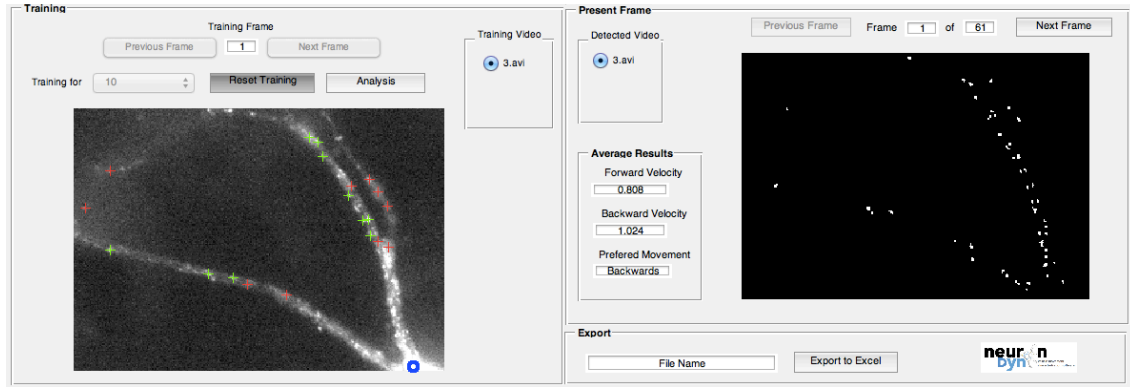
**Figure 6.12** – Intensity profile of an image from Dataset 3. a) Acquired image; B) Image produced after wiener filter; C) Image produced after Gaussian filter.

### *Detection and Segmentation*

The core of the tracking procedure is the vesicle detection. If this first step is accurate, the following steps will have better results. It is a difficult phase for some reasons such as: the recordings are often noisy; as an ANN classifier is used, if the training is not well performed, the detection will be erroneous.

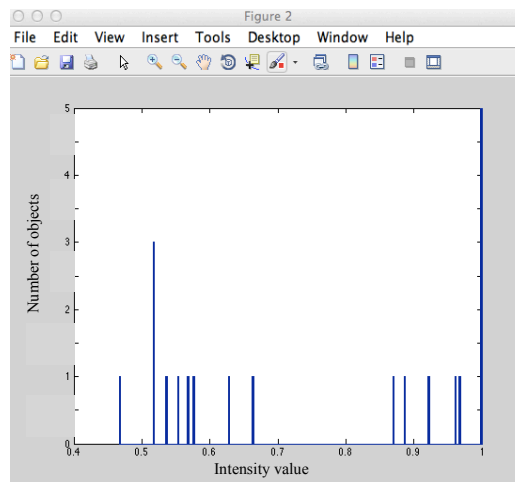
The developed algorithm performed well, even though the pre-processing steps

only consisted on a simple average filter, followed by a LoG (Laplacion of Gaussian – for high frequencies emphasis) enhancement window. Figure 6.13 shows the detected vesicles in *NeuronDyn* based in the training choices.



**Figure 6.13** – Detected vesicles in the first frame by *NeuronDyn* (right image), based in the training choices (left image). The training frame selected was the first one.

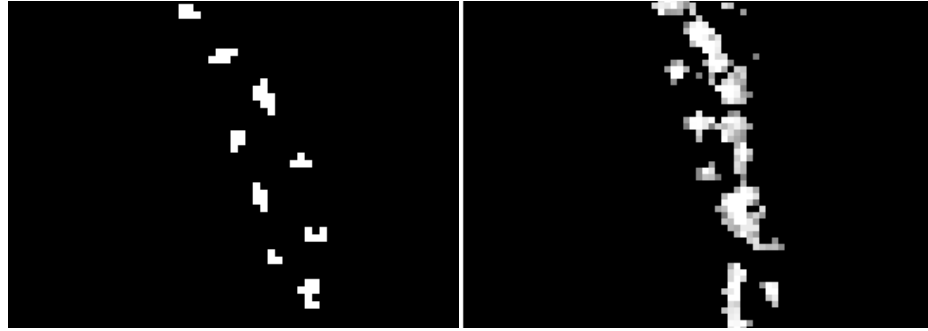
The automatic  $T$  parameter, used in equation 16 (see section 5.1.3), based in the intensity of the input values by the user showed to be a good implementation. If the user selects ambiguous vesicles with higher intensity value, the threshold will be higher, producing more restricted results with lower number of vesicles. As opposite, more vesicles (or noise) will be segmented. Figure 6.14 shows the histogram of the selected vesicles and non-vesicles for dataset 3, where the two classes can be well visualized, producing a threshold value in the valley.



**Figure 6.14** – Histogram with an example of the user's choices of 10 true vesicles (at right) and 10 ambiguous vesicles intensities (at left), for Dataset 3.

The vesicles segmentation produced by *NeuronDyn*, as also verified in *NeuronDynamics* does not contain the interference of background noise. Figure 6.15 shows the difference between the segmentation result by *NeuronDyn* and an example

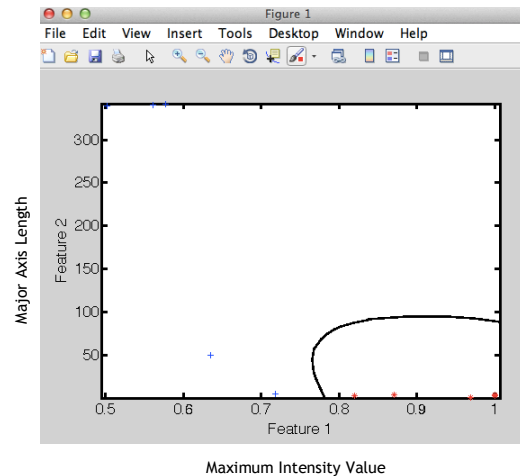
with interference.



**Figure 6.15** – Comparison of the vesicles segmentation. In the left, there is no interference of the background information on the segmentation, as seen in the right picture.

### Classification

The ANN classifier used showed to be a good solution, due to its flexibility and capacity of learning. Contrary to the *NeuronDynamics* classifier, which separates the data using a straight line, the ANN produces a flexible classification as shown in figure 6.16.

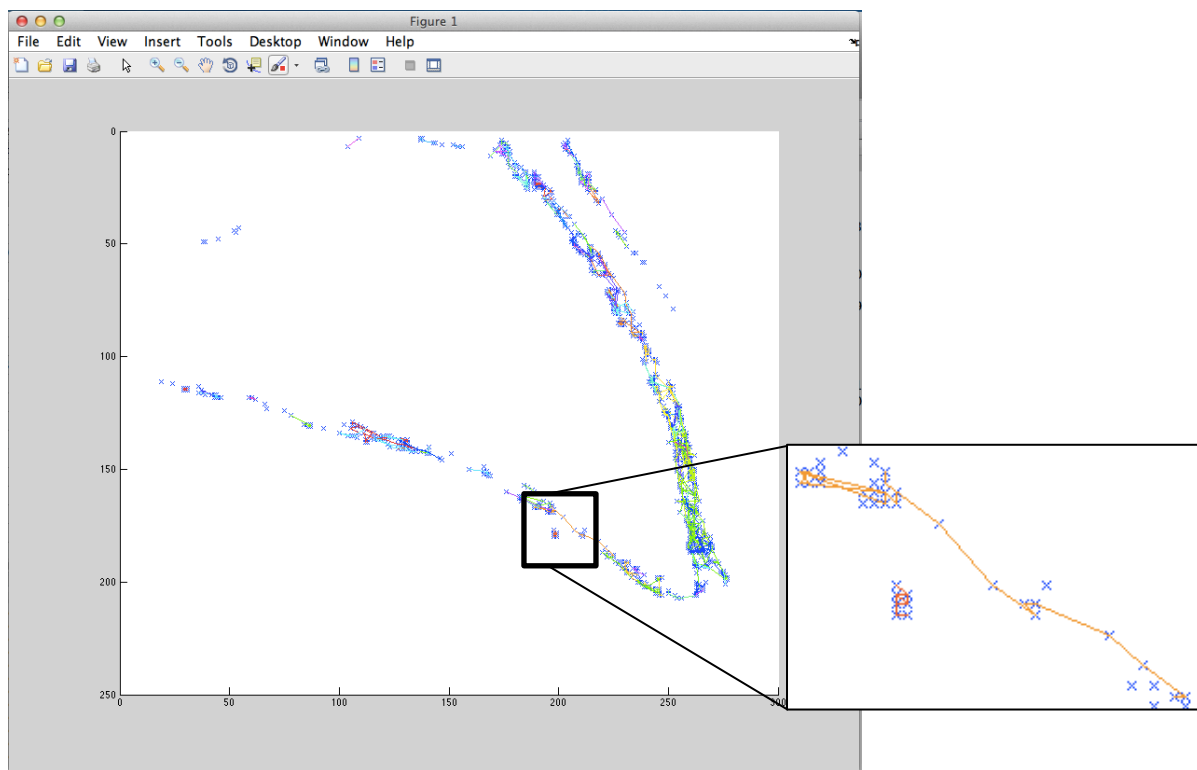


**Figure 6.16** – Classification performed by NeuronDyn between marked points, where feature 1 is the maximum intensity value, and feature 2 the major axis length. Real vesicles marked by the user are shown as red points and non-vesicles as blue points.

### Tracking

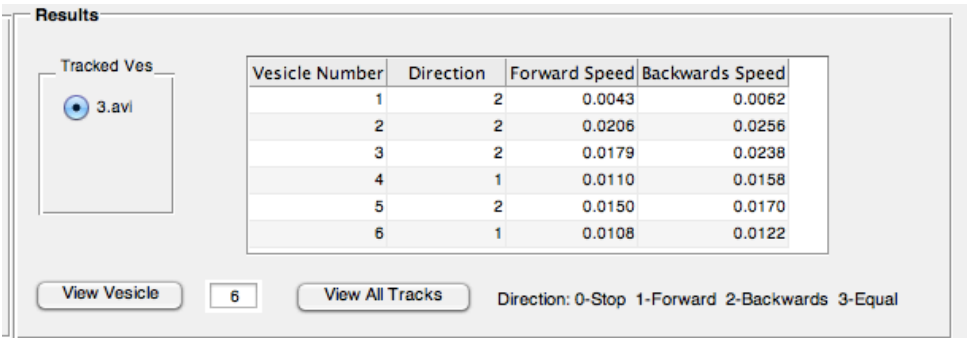
The principle behind tracking in *NeuronDyn* is a Global Nearest Neighbor, where the association is performed by Hungarian method. This method produces better results when compared with the Local Nearest Neighbor association, and considers some dynamic properties of the vesicles: birth and death, gaps 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

misdetected, continuing the track. Figure 6.17 shows all the detected vesicle and the tracks obtained in the Dataset 3 for the whole 61 frames. After the application of the parameters set by the user, the number of tracks decreases significantly.

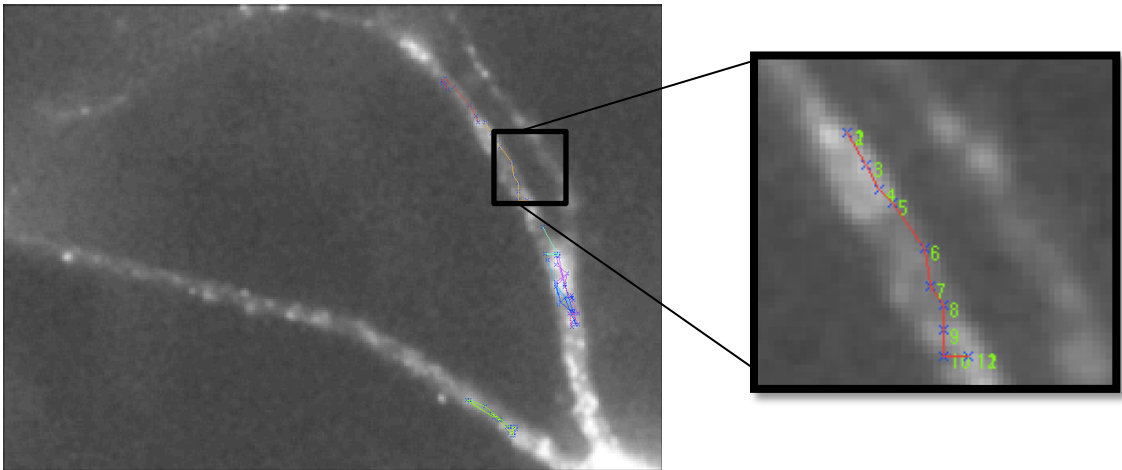


**Figure 6.17** – Detected vesicles and the tracks created to Dataset 3. The expansion of a portion of the image allows to verify the detected vesicles in that zone, and the tracks associated.

In *NeuronDyn* interface the tracking results can be observed in the Results section (figure 6.18), where all the vesicles tracked are presented in a table, with the main direction of movement and the speed resulted from the forward and backwards movements recorded. In the case of an analysis with more than one video, selecting other video will modify the results and dimension of the table, according to the tracks obtained. In this section it is also possible to visualize an image with all the tracks detected and an image with a specific vesicle and all its coordinates in the different frames. An example is shown in figure 6.19.



**Figure 6.18** – Tracking results section of NeuronDyn UI, with the tracks obtained for Dataset 3.



**Figure 6.19** – Image from “View all Tracks” option (left image) and a specific case of “View Vesicle” (right image), with the detected positions linked and a reference to the corresponding frame number.

The results are then exported to an Excel file, according to the selected parameters by the user. Figure 6.20 shows an example of the tracked vesicles for the first 10 frames of the Dataset 3. It is possible to visualize the date, selected parameters, preferred movement, forward and backwards velocities for each vesicle and the global values. The values for the velocities of the detected vesicles in the first frame are set to zero, being the measure started from that point. The first object does not present any movement during the analysis, which can be associated, for example, to a misdetection of a point that is not a vesicle.

As the preferred movement is forward, it is possible to conclude that the vesicles are mostly moving away from the cellular body, approaching the dendrites to pass the information.

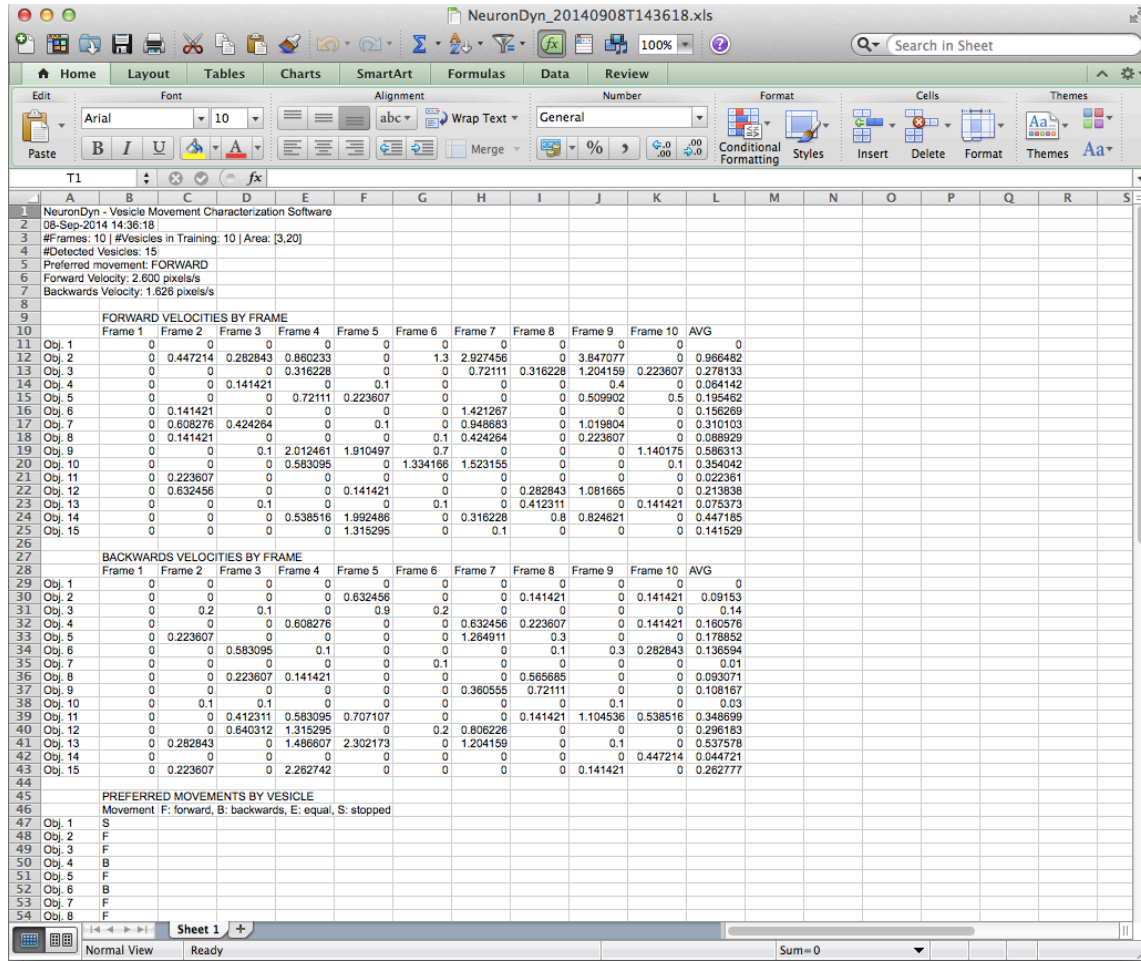


Figure 6.20 – Example of Excel results for Dataset 3, considering only 10 frames.

## Evaluation

The computer used for the evaluation of the segmentation in *NeuronDyn* was a Macintosh laptop with a processor Intel Core i5 with 1.3 GHz and 8GB RAM in MATLAB R2013a.

The results are shown in table 6.2. The values obtained show a good segmentation result, although the precision could be higher. This means that the algorithm is still detecting a few more points as vesicle than it should. Considering the *NeuronDynamics* and *FluoTracker* (See Section 5.2), the results showed near but lower values when compared with the ones obtained by *NeuronDynamics*. The non-linear classification implemented (ANN) did not presented an improvement in the results compared with the linear Bayesian Classification applied in *NeuronDynamics*. As the Neural Network consisted in a simple automatically trained feed-forward ANN classifier with one hidden layer it may have produced a network without optimal weights and or enough training.

Regarding the computational time of the algorithm, it shows that the robustness of

*NeuronDyn* produced a higher processing time when compared with the simple *NeuronDynamics*, although it is still much smaller than the other state of the art reference *FluoTracker*.

Table 6.2 – Evaluation of *NeuronDyn* Segmentation process.

<b><i>Specification</i></b>	<b><i>Value</i></b>
<b>Accuracy</b>	0,9987
<b>Sensitivity/Recall</b>	0,9227
<b>Specificity</b>	0,9990
<b>Precision</b>	0,7706
<b>F<sub>1</sub></b>	0,8398
<b>F<sub>0.5</sub></b>	5
<b>F<sub>2</sub></b>	1,25
<b>Computational Time [s]</b>	20.1796



## Chapter 7

### Conclusions

The advances in neuroscience and the increasing challenges in the neuronal transport allowed the development of solutions to analyse and characterize the particle's movement, however it continues to be undertaken 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. [83, 84] For these reasons *NeuronDyn* was developed as a customizable solution that allows a semi-automatic tracking of vesicles relying on information given and controlled by the user. This combines 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, miss detections and collisions of objects.

*NeuronDyn* is provided with an artificial neural network classifier training stage, which minimizes the influence of random errors. The algorithm “learns” from the user's experience, which, if the training is well performed, enables *NeuronDyn* to maintain accuracy while improving precision with faster computational time.

The main goals for *NeuronDyn* were well performed, which include:

- Possibility of choose the frame for the training;
- Global training process to a set of films obtained with the same parameters;

- Automatic and adaptable Threshold (T) parameter for each film;
- Division of the film by neuron's processes (axons and dendrites), performing the algorithm to the selected path;
- Robust tracking method that includes vesicle dynamic movements;
- Construction of an intuitive interface.

By eliminating the tedious process of manual tracking, this robust algorithm will add value to researchers work, being a tool to help their jobs.

### **Future Work**

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 classification and tracking algorithms and developing in a new programming language. Some of the tested videos had huge resolution and large number of frames, which ends with *Matlab*'s memory, making it impossible to analyse.

As this work will be continuous, in a future version, all the procedures should be evaluated to guarantee high performance of the algorithm.

More features from the vesicles can be extracted for the excel sheet, and a better UI should be constructed to show the individual tracks.

# References

- [1] Holzbaur, Erika LF. Axonal transport and neurodegenerative disease. "*Intracellular traffic and neurodegenerative disorders*". Springer Berlin Heidelberg, (2009): 27-39.
- [2] Zhang, Kai, Yasuko Osakada, Wenjun Xie, and Bianxiao Cui. "Automated image analysis for tracking cargo transport in axons." *Microscopy research and technique* 74, no. 7 (2011): 605-613.
- [3] Roy S, Zhang B, Lee VM, Trojanowski JQ., "Axonal transport defects: A common theme in neurodegenerative diseases". *Acta Neu-ropathol* (2005) 109:5–13.
- [4] De Vos, Kurt J., Andrew J. Grierson, Steven Ackerley, and Christopher CJ Miller. "Role of Axonal Transport in Neurodegenerative Diseases\*." *Annu. Rev. Neurosci.* 31 (2008): 151-173.
- [5] Morfini, Gerardo A., Matthew Burns, Lester I. Binder, Nicholas M. Kanaan, Nichole LaPointe, Daryl A. Bosco, Robert H. Brown et al. "Axonal transport defects in neurodegenerative diseases." *The Journal of Neuroscience* 29, no. 41 (2009): 12776-12786
- [6] Carpinteiro, Frederico A., Costa, Pedro M., Espinoza, Mario Sáenz, Silva, Ivo M., Cunha, João P. S., Neurondynamics: A Method for Neurotransmitter Vesicle Movement Characterization in Neurons; In Proceedings of the IEEE International Symposium on Biomedical Imaging; (2014)
- [7] Broeke, Jurjen HP, 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, no. 2 (2009): 378-384.
- [8] Kan, Andrey. "Automated analysis of time lapse microscopy images." (2012).
- [9] Asimov, Isaac, and Anthony Ravielli. *The human brain: Its capacities and functions*. Penguin, 1994.
- [10] Kandel, Eric R., James H. Schwartz, and Thomas M. Jessel. "Essentials of neural science and behavior." 1995.
- [11] Larsson, L-I., and J. F. Rehfeld. "Localization and molecular heterogeneity of cholecystokinin in the central and peripheral nervous system." *Brain Research* 165, no. 2 (1979): 201-218.
- [12] Asimov, Isaac, and Anthony Ravielli. *The human brain: Its capacities and functions*. Penguin, 1994.
- [13] Bekinshtein, Pedro, Martin Cammarota, Ivan Izquierdo, and Jorge H. Medina. "Reviews: BDNF and memory formation and storage." *The Neuroscientist* 14, no. 2 (2008): 147-156.
- [14] Eriksson, Peter S., Ekaterina Perfilieva, Thomas Björk-Eriksson, Ann-Marie Alborn, Claes Nordborg, Daniel A. Peterson, and Fred H. Gage. "Neurogenesis in the adult human hippocampus." *Nature medicine* 4, no. 11 (1998): 1313-1317.
- [15] Eriksson, Peter S., Ekaterina Perfilieva, Thomas Björk-Eriksson, Ann-Marie Alborn, Claes Nordborg, Daniel A. Peterson, and Fred H. Gage. "Neurogenesis in the adult human hippocampus." *Nature medicine* 4, no. 11 (1998): 1313-1317.
- [16] Farbman, Albert I. "Olfactory neurogenesis: genetic or environmental controls?." *Trends in neurosciences* 13, no. 9 (1990): 362-365.

- [17] Lois, Carlos, and Arturo Alvarez-Buylla. "Proliferating subventricular zone cells in the adult mammalian forebrain can differentiate into neurons and glia." *Proceedings of the National Academy of Sciences* 90, no. 5 (1993): 2074-2077.
- [18] Johnston, Daniel, Jeffrey C. Magee, Costa M. Colbert, and Brian R. Christie. "Active properties of neuronal dendrites." *Annual review of neuroscience* 19, no. 1 (1996): 165-186.
- [19] Neuron anatomy.  
[http://classconnection.s3.amazonaws.com/93/flashcards/1278093/jpg/neuron\\_structure1353912562903.jpg](http://classconnection.s3.amazonaws.com/93/flashcards/1278093/jpg/neuron_structure1353912562903.jpg) last checked: 12/09/2014
- [20] Kandel, Eric R., James H. Schwartz, and Thomas M. Jessell, eds. *Principles of neural science*. Vol. 4. New York: McGraw-Hill, 2000.
- [21] Stuart, Greg, Nelson Spruston, and Michael Häusser, eds. *Dendrites*. Oxford University Press, 2007.
- [22] Fatt, Paul. "Sequence of events in synaptic activation of a motoneurone." *Journal of Neurophysiology* 20, no. 1 (1957): 61-80.
- [23] Häusser, Michael, and Bartlett Mel. "Dendrites: bug or feature?." *Current opinion in neurobiology* 13, no. 3 (2003): 372-383.
- [24] Craig, Ann Marie, and Gary Banker. "Neuronal polarity." *Annual review of neuroscience* 17, no. 1 (1994): 267-310.
- [25] Gerstner, Wulfram, and Werner M. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.
- [26] Brady, Scott T. "A novel brain ATPase with properties expected for the fast axonal transport motor." (1985): 73-75.
- [27] Dayan, Peter, Laurence F. Abbott, and L. Abbott. "Theoretical neuroscience: Computational and mathematical modeling of neural systems." 2001.
- [28] Raimondi, Andrea, Shawn M. Ferguson, Xuelin Lou, Moritz Armbruster, Summer Paradise, Silvia Giovedi, Mirko Messa et al. "Overlapping role of dynamin isoforms in synaptic vesicle endocytosis." *Neuron* 70, no. 6 (2011): 1100-1114
- [29] Shepherd, Gordon M., ed. *The synaptic organization of the brain*. Vol. 4. Oxford: Oxford University Press, 2004.
- [30] Johnston, M. F., S. A. Simon, and F. Ramon. "Interaction of anaesthetics with electrical synapses." (1980): 498-500.
- [31] Hille, Bertil. *Ion channels of excitable membranes*. Vol. 507. Sunderland, MA: Sinauer, 2001.
- [32] Rudnick, Gary. "Bioenergetics of neurotransmitter transport." *Journal of bioenergetics and biomembranes* 30, no. 2 (1998): 173-185.
- [33] Südhof, Thomas C. "The synaptic vesicle cycle." *Annu. Rev. Neurosci.* 27 (2004): 509-547
- [34] Holzbaur EL., Motor neurons rely on motor proteins. *Trends Cell Biol* (2004) 14:233–240.
- [35] Welte, Michael A. "Bidirectional transport along microtubules." *Current Biology* 14, no. 13 (2004): R525-R537.
- [36] Vale, Ronald D. "Intracellular transport using microtubule-based motors." *Annual review of cell biology* 3, no. 1 (1987): 347-378.
- [37] Conde, Cecilia, and Alfredo Cáceres. "Microtubule assembly, organization and dynamics in axons and dendrites." *Nature Reviews Neuroscience* 10, no. 5 (2009): 319-332.
- [38] Goldstein, Lawrence SB, and Alastair Valentine Philp. "The Road Less Traveled 1: Emerging Principles of Kinesin Motor Utilization." *Annual review of cell and developmental biology* 15, no. 1 (1999): 141-183.

- [39] Morfini, Gerardo A., Matthew R. Burns, David L. Stenoien, and Scott T. Brady. "Axonal transport." *Basic Neurochemistry: Principles of Molecular, Cellular, and Medical Neurobiology* (2011): 146-164.
- [40] Barkus, Rosemarie V., Olga Klyachko, Dai Horiuchi, Barry J. Dickson, and William M. Saxton. "Identification of an axonal kinesin-3 motor for fast anterograde vesicle transport that facilitates retrograde transport of neuropeptides." *Molecular biology of the cell* 19, no. 1 (2008): 274-283.
- [41] Boldogh, Istvan R., and Liza A. Pon. "Mitochondria on the move." *Trends in cell biology* 17, no. 10 (2007): 502-510.
- [42] Hollenbeck, Peter J., and William M. Saxton. "The axonal transport of mitochondria." *Journal of cell science* 118, no. 23 (2005): 5411-5419.
- [43] Levy, Jennifer R., and Erika LF Holzbaur. "Cytoplasmic dynein/dynactin function and dysfunction in motor neurons." *International journal of developmental neuroscience* 24, no. 2 (2006): 103-111.
- [44] Deacon, Sean W., Anna S. Serpinskaya, Patricia S. Vaughan, Monica Lopez Fanarraga, Isabelle Vernos, Kevin T. Vaughan, and Vladimir I. Gelfand. "Dynactin is required for bidirectional organelle transport." *The Journal of cell biology* 160, no. 3 (2003): 297-301.
- [45] Hirokawa, Nobutaka, and Reiko Takemura. "Molecular motors and mechanisms of directional transport in neurons." *Nature Reviews Neuroscience* 6, no. 3 (2005): 201-214.
- [46] Oztas, Emin. "Neuronal tracing." *Neuroanatomy* 2 (2003): 2-5.
- [47] Bospoort, R. Van de., Molecular mechanisms of neuronal dense core vesicle release. Vrije Universiteit Amsterdam. Doctoral Thesis (2013).
- [48] Kim, Taeyoon, Marjorie C. Gondré-Lewis, Irina Arnaoutova, and Y. Peng Loh. "Dense-core secretory granule biogenesis." *Physiology* 21, no. 2 (2006): 124-133.
- [49] Li, Jia Yi, and Annica Dahlström. "Axonal transport of neuropeptides: Retrograde tracing study in live cell cultures of rat sympathetic cervical ganglia." *Journal of neuroscience research* 85, no. 12 (2007): 2538-2545.
- [50] Chevalier-Larsen, Erica, and Erika LF Holzbaur. "Axonal transport and neurodegenerative disease." *Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease* 1762, no. 11 (2006): 1094-1108.
- [51] Gauthier, Laurent R., Bénédicte C. Charrin, Maria Borrell-Pagès, Jim P. Dompierre, Hélène Rangone, Fabrice P. Cordelières, Jan De Mey et al. "Huntingtin controls neurotrophic support and survival of neurons by enhancing BDNF vesicular transport along microtubules." *Cell* 118, no. 1 (2004): 127-138.
- [52] Simic, Goran, et al. "Volume and number of neurons of the human hippocampal formation in normal aging and Alzheimer's disease." *Journal of Comparative Neurology* 379.4 (1997): 482-494.
- [53] Stokin, Gorazd B., and Lawrence SB Goldstein. "Axonal transport and Alzheimer's disease." *Annu. Rev. Biochem.* 75 (2006): 607-627.
- [54] Schindowski, K., K. Belarbi, and L. Buee. "Neurotrophic factors in Alzheimer's disease: role of axonal transport." *Genes, Brain and Behavior* 7, no. s1 (2008): 43-56.
- [55] Staff, Nathan P., Eduardo E. Benarroch, and Christopher J. Klein. "Neuronal intracellular transport and neurodegenerative disease." *Neurology* 76, no. 11 (2011): 1015-1020.
- [56] Grinvald, Amiram, and Rina Hildesheim. "VSDI: a new era in functional imaging of cortical dynamics." *Nature Reviews Neuroscience* 5.11 (2004): 874-885.
- [57] Tsien RY (2003). Imagining imaging's future. *Nature Reviews Molecular Cell Biology* 4, S16–S21

- [58] Bozzola, John J., and Lonnie Dee Russell. *Electron microscopy: principles and techniques for biologists*. Jones & Bartlett Learning, 1999.
- [59] Kankaanpää, Pasi. "Developing Bioimage Informatics—from Microscopy to Software Solutions—with alpha2beta1 Integrin as a Case Study." (2014).
- [60] Pawley, James B., ed. *Handbook of biological confocal microscopy*. Springer, 1995.
- [61] Differences between CLSM and Spinning Disk Confocal Microscopy  
[http://o.quizlet.com/ditTiwXSC11b0eyd7HsY7w\\_m.png](http://o.quizlet.com/ditTiwXSC11b0eyd7HsY7w_m.png) Last checked: 12/09/2014
- [62] Vonesch, Cédric, François Aguet, J-L. Vonesch, and Michael Unser. "The colored revolution of bioimaging." *Signal processing magazine, IEEE* 23, no. 3 (2006): 20-31.
- [63] Rangayyan, Rangaraj M. *Biomedical image analysis*. CRC press, 2004.
- [64] Narkhede, H. P. "Review of Image Segmentation Techniques." *International Journal of Science and Modern Engineering* 1, no. 8 (2013): 54-61.
- [65] Gonzalez, Rafael C., Richard E. Woods, and Steven L. Eddins. *Digital image processing using MATLAB*. Vol. 2. Knoxville: Gatesmark Publishing, 2009.
- [66] Ahmadi, Reza, Javad Kangarani Farahani, Farbod Sotudeh, Ashkan Zhaleh, and Saeid Garshasbi. "Survey of Image Denoising Techniques." *Life Science Journal* 10, no. 1 (2013).
- [67] Saxena, Chandrika, and Deepak Kourav. "Noises and Image Denoising Techniques: A Brief Survey." (2014)
- [68] Dubey, Mrs Deepty. "A Survey on Image Noises and Denoise Techniques." (2012).
- [69] Yu, Jinhua, and Jinglu Tan. "Object density-based image segmentation and its applications in biomedical image analysis." *Computer methods and programs in biomedicine* 96, no. 3 (2009): 193-204.
- [70] Maini, Raman, and Himanshu Aggarwal. "A comprehensive review of image enhancement techniques." *arXiv preprint arXiv:1003.4053* (2010).
- [71] Erdt, Marius, Sebastian Steger, and Georgios Sakas. "Regmentation: A new view of image segmentation and registration." *Journal of Radiation Oncology Informatics* 4, no. 1 (2012): 1-23.
- [72] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." *Automatica* 11, no. 285-296 (1975): 23-27.
- [73] Fu, King-Sun, and J. K. Mui. "A survey on image segmentation." *Pattern recognition* 13, no. 1 (1981): 3-16.
- [74] Dey, Vivek, Y. Zhang, and M. Zhong. *A review on image segmentation techniques with remote sensing perspective*. na, 2010.
- [75] Adams, Rolf, and Leanne Bischof. "Seeded region growing." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16, no. 6 (1994): 641-647.
- [76] Region Growing Example [http://www.cs.cf.ac.uk/Dave/Vision\\_lecture/region\\_growing.gif](http://www.cs.cf.ac.uk/Dave/Vision_lecture/region_growing.gif) last checked: 12/09/2014
- [77] Kaur, Dilpreet, and Yadwinder Kaur. "Various Image Segmentation Techniques: A Review." (2014).
- [78] Zeng, Qingbing, Shiping Wang, Yubin Miao, and Chengliang Liu. "Algorithm based on marker-controlled watershed transform for overlapping plant fruit segmentation." *Optical Engineering* 48, no. 2 (2009): 027201-027201.
- [79] Example of snake application, present in [www.cs.uu.nl/docs/vakken/ibv/reader/chapter10.pdf](http://www.cs.uu.nl/docs/vakken/ibv/reader/chapter10.pdf) last checked 12/09/2014
- [80] Khan, A. M., Ravi, S. "Image Segmentation Methods: A Comparative Study". *International Journal of Soft Computing and Engineering* ISSN: 2231-2307, Volume 3, Issue 4 (2013).

- [81] Wang, Wei Xing. "Binary image segmentation of aggregates based on polygonal approximation and classification of concavities." *Pattern Recognition* 31, no. 10 (1998): 1503-1524.
- [82] Ping Tian, Dong. "A Review on Image Feature Extraction and Representation Techniques." *International Journal of Multimedia and Ubiquitous Engineering* (2013).
- [83] Shape descriptors  
[http://users.monash.edu.au/~dengs/resource/papers/thesis\\_abstract\\_files/image003.jpg](http://users.monash.edu.au/~dengs/resource/papers/thesis_abstract_files/image003.jpg) last checked 12/09/2014
- [84] Lalitha, M., M. Kiruthiga, and C. Loganathan. "A Survey on Image Segmentation through Clustering Algorithm." *International Journal* (2013).
- [85] Kanungo, Tapas, 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, no. 7 (2002): 881-892.
- [86] Mohamed, Nevin A., M. N. Ahmed, and A. Farag. "Modified fuzzy c-mean in medical image segmentation." In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 6, pp. 3429-3432. IEEE, 1999.
- [87] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24, no. 5 (2002): 603-619.
- [88] Machine learning history. <http://www.erogol.com/wp-content/uploads/2014/05/test.jpg> last checked 12/09/2014
- [89] Rickard, Heather Erin. "Feature selection for self-organizing feature map neural networks with applications in medical image segmentation." PhD diss., University of Louisville, 2001.
- [90] Domingos, Pedro, and Michael Pazzani. "On the optimality of the simple Bayesian classifier under zero-one loss." *Machine learning* 29, no. 2-3 (1997): 103-130.
- [91] Support vector machines <http://www.mathworks.com/help/stats/svmhyperplane.png> last checked 12/09/2014
- [92] Milan, Anton. "Energy Minimization for Multiple Object Tracking." (2013)
- [93] Object tracking approaches  
<http://www.wu.ece.ufl.edu/courses/eel6562f07/Taxonomy%20of%20tracking%20methods.gif> last checked 12/09/2014
- [94] Patel, Sandeep Kumar, and Agya Mishra. "Moving Object Tracking Techniques: A Critical Review." *Indian Journal of Computer Science & Engineering* 4, no. 2 (2013).
- [95] Mourllion, Benjamin, Dominique Gruyer, Cyril Royere, and Sébastien Thérout. "Multi-hypotheses tracking algorithm based on the belief theory." In *Information Fusion, 2005 8th International Conference on*, vol. 2, pp. 8-pp. IEEE, 2005.
- [96] Rout, Rupesh Kumar. "A survey on object detection and tracking algorithms." PhD diss., 2013.
- [97] Goshima, Yoshio, 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, no. 3 (2012): 3414-3430.
- [98] Coleman, Paul D., Catherine F. Garvey, John H. Young, and William Simon. "Semiautomatic tracking of neuronal processes." In *Computer Analysis of Neuronal Structures*, pp. 91-109. Springer US, 1977.
- [99] Oh, Songhwai, Stuart Russell, and Shankar Sastry. "Markov chain Monte Carlo data association for general multiple-target tracking problems." In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 1, pp. 735-742. IEEE, 2004.

- [100] Rodriguez, Alfredo, Douglas B. Ehlenberger, Dara L. Dickstein, Patrick R. Hof, and Susan L. Wearne. "Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images." *PLoS One* 3, no. 4 (2008): e1997.
- [101] Meijering, Erik, Oleh Dzyubachyk, and Ihor Smal. "Methods for cell and particle tracking." *Methods Enzymol* 504, no. 9 (2012): 183-200.
- [102] Beltman, Joost B., Athanasios FM Marée, and Rob J. de Boer. "Analysing immune cell migration." *Nature Reviews Immunology* 9, no. 11 (2009): 789-798.
- [103] Bakal, Chris, John Aach, George Church, and Norbert Perrimon. "Quantitative morphological signatures define local signaling networks regulating cell morphology." *science* 316, no. 5832 (2007): 1753-1756.
- [104] Smachina, C. , "Image Processing Techniques and Segmentation Evaluation", Doctoral Thesis, Technical University "Gheorghe Asachi" (2011).
- [105] Bradley, Andrew P. "The use of the area under the ROC curve in the evaluation of machine learning algorithms." *Pattern recognition* 30, no. 7 (1997): 1145-1159.
- [106] Yau, H. T., Y. K. Lin, L. S. Tsou, and C. Y. Lee. "An adaptive region growing method to segment inferior alveolar nerve canal from 3d medical images for dental implant surgery." *Computer-Aided Design and Applications* 5, no. 5 (2008): 743-752.
- [107] A. Khodjakov and C. Rieder, "Imaging the division process in living tissue culture cells," *Methods*, vol. 38, no. 1, pp. 2–16, 2006.

# Appendix



# **Appendix A**

## **Cell Tracking Challenge**



To obtain practice in the addressed problem of the thesis and as a training step to subsequently design *NeuronDyn*, an automatic cell tracking algorithm was created. After a paper acceptance in the IEEE International Symposium on Biomedical Imaging (ISBI) 2014, a participation on the Second Cell Tracking Challenge promoted by ISBI was made. The proposed challenge intended to produce algorithms to help researchers in the cell's quantification, a central topic in biomedical engineering, which deals with situations of touching cells, unions and divisions, phenomena also observed in vesicles.

**Clinical relevance.** Cell motility is an important area in biomedical research. The ability of cells to exert forces on their environment and alter their shape as they move is essential for various biological processes such as the immune response, embryonic development, or tumorigenesis. Recent technological advances in confocal fluorescence microscopy gave researchers the opportunity to examine these processes in three dimensions within a living organism. It requires proper detection of both movement and morphological changes that the cell suffers as it moves through the environment. Manual cell tracking is an extremely laborious task, due to the large amount of image data acquired during live-cell studies. Thus, the analysis of time-lapse experiments increasingly relies on automated image processing techniques.

## **Methodology**

### *Image Acquisition*

The dataset videos used were provided by the Cell Tracking Challenge. The created algorithm was only tested with four videos, with varying resolution, frame rate and number of frames. The cell density and noise levels varied significantly between the videos. The ground truth results were manually annotated by three experts from different institutions. The used datasets are presented in figure A.1.

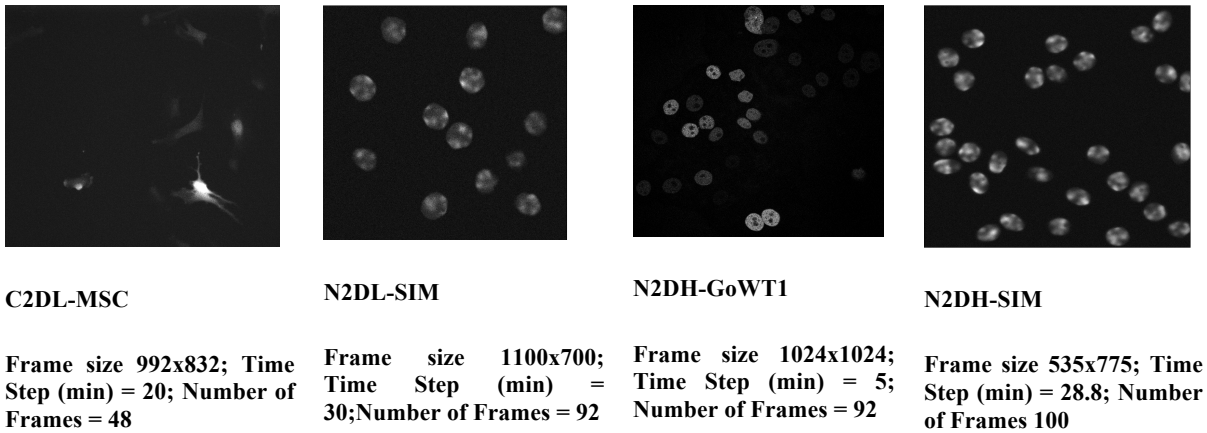


Figure A.1 – Datasets used in the Cell Tracking Challenge

*Cell Tracking Algorithm*

As one of the requirements was to produce an algorithm with low computational cost, the considered methods for the cell tracking algorithm developed were simple, fast and easy to implement. The main steps of the designed method were Image Denoising, Enhancement, Segmentation, Feature Extraction and Tracking, producing as output an .txt file with all detected cells, its start and end frame and the cell from which they came (in the case of cells resulting from divisions or unions).

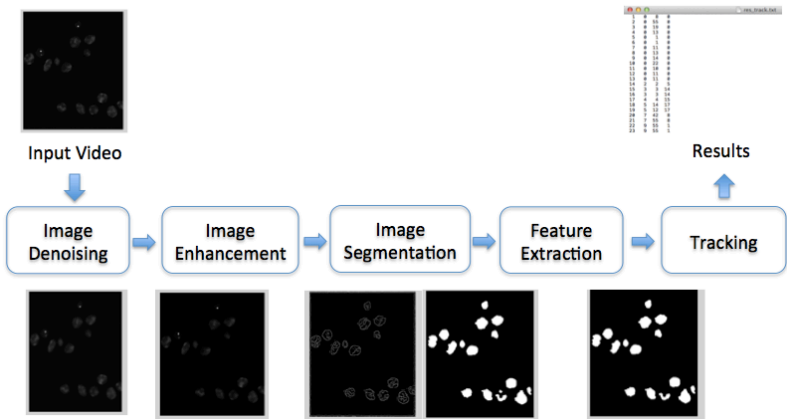


Figure A.2 – Overview of the Cell Tracking Algorithm

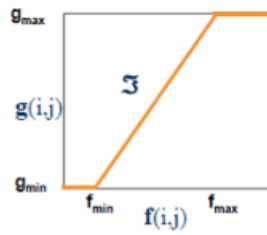
*Image Denoising and Enhancement*

To reduce the presence of noise verified in the videos a denoising stage was performed, to subsequently produce better results in the segmentation process.

The noise presented in the images had a Gaussian distribution, and the solution applied was an adaptive wiener filter. This approach allowed smoothing the image in

function of the local variance (where higher variance produced lower smoothing and smaller variance, additional smoothing). An adaptive filter obtains better results than comparable linear filtering, preserving better the edges and other high-frequency parts of the image.

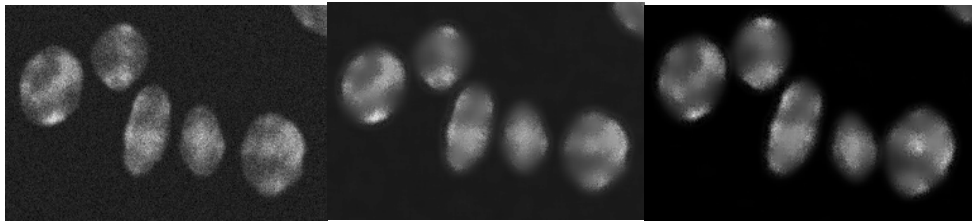
After this step, a simple linear expansion of the histogram was performed applying the equation 19 (which is explained in figure A.3), in order to enhance the cells in the image.



**Figure A.3 – Linear expansion of the histogram.**

$$g = \frac{g_{\max} - g_{\min}}{f_{\max} - f_{\min}} (f - f_{\min}) + g_{\min} \quad (19)$$

Where  $g(i,j)$  is the new image and  $f(i,j)$  is the initial image.  $g_{\max}$  and  $g_{\min}$  are the new higher and lower values, and  $f_{\max}$  and  $f_{\min}$  are the older higher and lower values, respectively. Figure A.4 shows the result of the two mentioned filters.



**Figure A.4 - Example of a) initial image, b) after denoising and c) after enhancement**

### *Image Segmentation*

As the images resulting from the previous step produced well-differentiated cells from the background, a simple image based segmentation technique was used. By implementing a boundary/edge-based method like Canny operator, a sensible operator that identifies the abrupt changes and suppresses the constant gray-level areas, it was

possible to obtain well-segmented cells. A illustration of a Canny segmentation is shown in figure A.5.

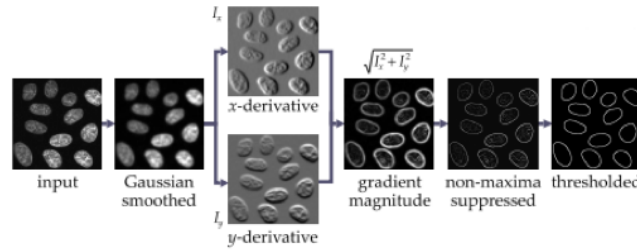


Figure A.5 – Illustration of Canny Edge segmentation.

The obtained cells from the Canny segmentation process were then subjected to an edge linking to assure that all edges were properly connected. The edges were filled to obtain correct cell segmentation and then a set of morphological operators was added (close and open operators) to guarantee that different cells were not connecting to each other.

At this point, the objects touching the limits of the video (in width and height) were removed from the search by changing its value to zero. These cells were entering and exiting in different frames, being later labelled as different cells. An example of the obtained results of Segmentation is shown in figure A.6.



Figure A.6 - Cells after a) Enhancement method, b) Canny segmentation, c) Edge linking and morphological operators.

### *Feature Extraction and Tracking*

From the segmentation process some noisy points were classified as objects of interest. To solve this problem, the area of all candidates was extracted and used as factor to differentiate cells and noise. When the data sets have large number of cells (more than 20), the variance of area between them is small which makes it possible to eliminate objects with area below 100 pixels<sup>2</sup>. When the number of cells is smaller than 20, the variance value can be higher and only objects with smaller area than 20 pixels<sup>2</sup> are eliminated (being classified as noise). From the final detected objects the area and the centroid positions were then extracted.

The tracking system applied was a modified Nearest-Neighbor Association. For each object the following characteristics were computed in each frame: Centroid position, Centroid position in the last frame, Actual Area, Last Area, Cell index, the frame where the cell entered and ended, and Last track (cell index from which it came). In the case of unions, the previous characteristics of the new cell were passed from the cell with higher area.

For the position of each object in each frame, the distance to all cells in the last frame was measured. If the minimum distance obtained is lower than 15 pixels the association is performed and the start, end frame and last track is not altered. In the opposite case, minimum distance is higher than 15 pixels and new possibilities are assumed: The area of the associated cells (between different frames) is compared and if the new area is higher than  $1.33 \cdot \text{LastArea}$  a Merging/Union situation is assumed, and the characteristics of the new object are changed as also from the last object. If the Area is lower than  $0.67 \cdot \text{LastArea}$  a split/division occurs, and if it stays between  $0.67 \cdot \text{LastArea}$  and  $1.33 \cdot \text{LastArea}$ , the number of cells in the two consecutive frames is obtained, as well as the number of divisions and unions, which allows to conclude if the cell continues or if it's a new one.

### Results and Discussion

The labelled cells from the tracking process were then exported to a .txt file, where each cell was represented by its index, the initial and end frame and the last track. Also, each frame was saved as an image, with all the labels in the cells. Figure A.7 shows an example of an image with labelled cells and the results obtained in the .txt file.

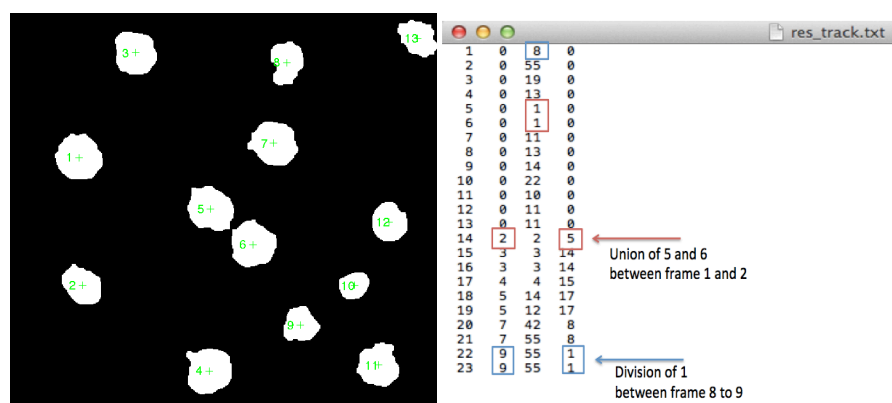


Figure A.7 – Example of image obtained and the .txt of the data set.

Two examples of division and union detections are presented in figure A.8 and A.9. In the first, a cell is divided in two, from frame 8 to 9, and two different labels are

assigned. In the second case cells 5 and 6 are merged in cell 14, from frame 6 to 7, also perceptible in the excel sheet.

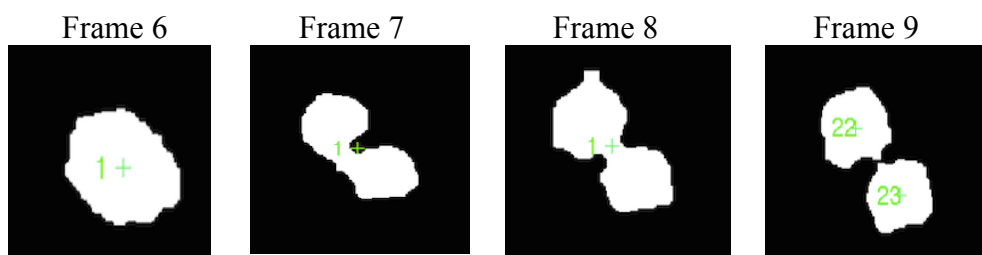


Figure A.8 – Example of division detected by the algorithm.

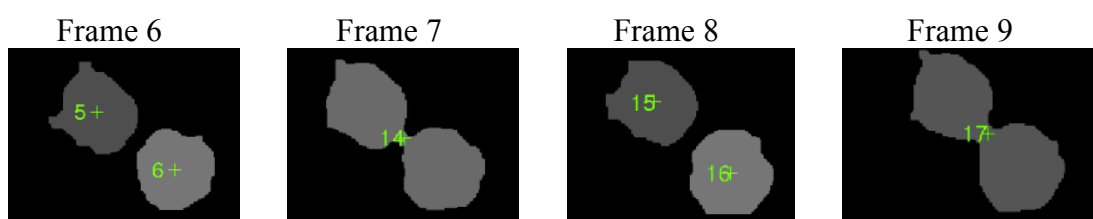
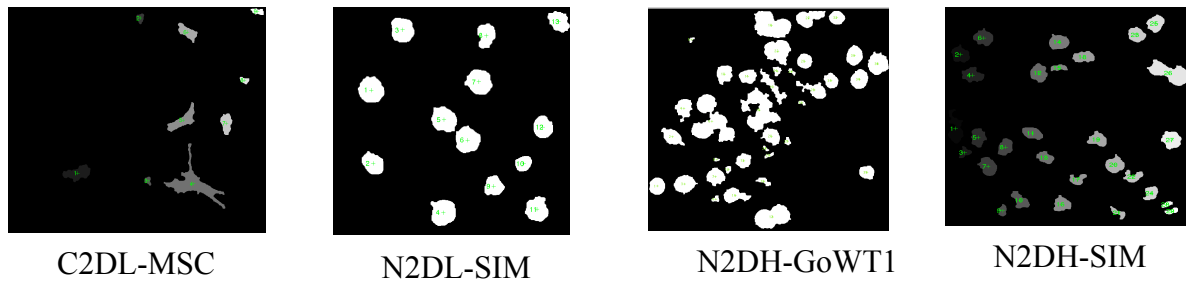


Figure A.9 – Example of merge marked by the algorithm.

Although the algorithm correctly labelled the cells tacking in consideration the segmentation result, this one did not performed always properly. Cells 5 and 6 in the last example were moving closely, but in some frames the algorithm segmented them as only one object, producing bad results in the tracking association (constantly merging and dividing the cells, changing its index). This error allows concluding that each step from image processing and analysis algorithms needs to be as more precise as possible to obtain good results in the tracking process.

The touching/near cells problem was not solved and the algorithm performed poorly in those situations. The tracking process implemented showed to be a good and simple solution. Other examples of the segmentation results are shown in the figure A.10, presenting three good segmentation results and one image with lower precision.

Other important feature that could have been used is the eccentricity. This feature would allow to differentiate cells that were having division/fusion from the ones that were not, since the cell shape changes significantly between those processes.



**Figure A.10 – Results from segmentation and labelling.**

### *Conclusions*

Cell tracking is one of the most important and common tasks for biomedical researchers, however it continues to be achieved manually. Precise automatic tracking systems are a challenging problem in computer vision and digital image processing. There are a large number of image/video properties that increase the difficulty of this task, such as background noise, resolution, frame rate, among others. Also, due to the large number of cell types with different features and behaviours, designing a universal cell tracking system is impractical. So, a semi-automatic approach has the advantage of the user to control cell segmentation/detection process, being expected to have best results.



## **Appendix B**

### **CellREM: The Recursive Error Metric For Cell Tracking**



From the discussion of the work developed in the Cell Tracking Challenge, where the goal was to obtain a precise and fast cell-tracking algorithm, an idea to create a new metric for cell tracking emerged. The proposed recursive error calculation metric (*CellREM*) minimizes mismatches during tracking. This metric is based on the objects properties along consecutive frames - distance between centroids, shape factor and eccentricity - and was designed to be as flexible as possible, allowing the user to adjust the weights and increasing the system's performance during tracking. Furthermore, it can also be implemented in already available tracking algorithms or be used as a benchmark for future algorithms tracking performance evaluation. This work produced a publication paper in the IEEE PGBiomed 2014 Conference at Warwick university where the author did a poster presentation.

**Clinical relevance.** Automatic object tracking is a challenging problem in biomedical image processing, and a need to study many important biological processes. Consequently, associated to very high temporal and spatial resolution imaging techniques, good object segmentation and tracking are of paramount importance to obtain results that not only speed up the analysis process, but also give way to high accuracy and precision, that are difficult to achieve by manual tracking, a user-biased, intensive and tedious labor procedure.

In general terms tracking consists in assigning each object of interest (OOI) in the present  $k$  frame and matching to the OOI in the next  $k+1$  frame.

Several commercial automatic tracking programs have been developed over the last decades — such as *ImageJ*, *CellTracker* and *MetaMorph* — that detect the individual positions at each frame and link them over time, presenting different degrees of accuracy and complexity.

When the number of objects is the same through the film, simple tracking functions such as the nearest neighbour association (NNA) can be applied, where each object is associated with the closest possible position in the next frame. Its advantages are related to low computation cost and simplicity, allowing to quickly track objects that are well individualized and do not change its position abruptly. [6]

In fact, the frame rate is an important parameter in the tracking performance. Higher frame rates are normally associated with better results using the NNA approach, although increasing the frame rate arbitrarily can compromise the health of the cells by phototoxicity. On the other hand, low frame rates result in large displacements of objects that produce poor results with the NNA. [6,107]

In an effort to solve these problems and maintain the simplicity and computational cost of the NNA approach, other cell characteristics apart of distance between centroids should be considered, such as the shape factor and eccentricity. In order to address these

features to a simple NNA approach this method — named *CellREM* — was developed, based on a recursive error metric.

## Methodology

### *CellREM*

The recursive error metric *CellREM* was developed (using *Matlab*) to substitute/complement the simple NNA commonly used in many biomedical image processing problems. In an effort to upgrade and improve the system's tracking, the matching function was enhanced by considering not only the distance between centroids but also the shape factor and eccentricity, as shown in Equation 20.

$$E = \frac{\alpha D + \beta(\Delta f_c) + \gamma(\Delta e)}{\alpha + \beta + \gamma} \quad (20)$$

Where  $\alpha$ ,  $\beta$  and  $\gamma$  are weighting factors for distance (D), shape factor (fc) and eccentricity (e). Their complete description can be found in Equations 21a, 21b and 21c.

$$D = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \quad (21a)$$

$$\Delta f_c = |f_{c_{k+1}} - f_{c_k}| = \left| \frac{4\pi A_{k+1}}{P_{k+1}^2} - \frac{4\pi A_k}{P_k^2} \right| \quad (21b)$$

$$\Delta e = |e_{k+1} - e_k| = \left| \frac{L_{k+1} - l_{k+1}}{L_{k+1} + l_{k+1}} - \frac{L_k - l_k}{L_k + l_k} \right| \quad (21c)$$

Where D is a 2D Euclidean distance (in pixels) normalized to the farthest next-frame centroid.  $f_c$  is a circularity shape factor—which relates area (A) and perimeter (P) into one single metric—and is also normalized to the largest difference in shape factor between all the possible next frame (k+1) objects. In the same way, e (eccentricity) relates the major axis (L) with the minor axis (l) of the objects, and is also normalized to the most different one. The orbital eccentricity definition (e in Equation 21c) was used so  $-1 \leq e \leq 1$ , which allowed to also take into consideration the orientation of the objects in the frame.

The shape factor and the eccentricity allowed to relate several cells properties in a comprehensive, yet simple way. The normalizations of D,  $f_c$  and e subsequently allowed the creation of a standard error metric that can be used not only for the purposes of the present project, but also for different segmentation scenarios or other tracking algorithms.

The nearest neighbour association (NNA) simply uses Equation 21a to associate the nearest centroid of the region of interest in the k frame to the next k + 1 one. Its the

same as using the *CellREM* combination of parameters:  $\alpha = 1$ ,  $\beta = 0$  and  $\gamma = 0$ . *CellREM*, on the other hand is a flexible algorithm that can be tuned to obtain the best combination of parameters to perform the tracking.

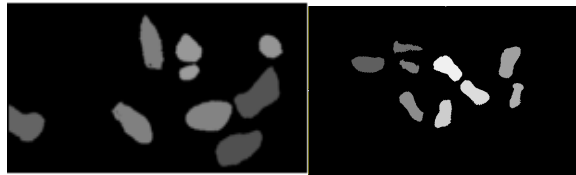
Some meaningful quantitative measures related to the motility, diffusivity, velocity and morphology of the moving objects could be obtained from the CellREM. For example, the object's speed is estimated by dividing the distance travelled (in pixels) by the video frame rate (in frames/second).

#### *Dataset Characterization*

The data used for this work was composed of two videos: Dataset 1, with a resolution of 160x82 pixels<sup>2</sup> and frame rate of 10 frames per second (fps) (51 frames in 5,1 seconds); Dataset 2, with 640x400 pixels<sup>2</sup> and frame rate of 6 fps (33 frames in 5,5 seconds). The two datasets were composed of simulated nuclei moving on a flat surface (N2DH-SIM cells), with a pixel size of 0.125 x 0.125 micrometers, using Cytopacq to simulate the data. The videos displayed fluorescently labeled nuclei of the HL60 cell line migrating on a flat 2D surface. The objective lens/numerical aperture used was 40x plan-apchromat with 1.3 oil.

## **Results and Discussion**

In this research, CellREM and NNA were applied to two datasets of the same type of cells (N2DH-SIM cells). The image sequences used to implement the two tracking algorithms were the ground truth results of the segmentation process (figure B.1), where a constant number of cells (9) was considered in the two films.



**Figure B.1 - At left Data set 1 with 160x82 pixels, and at right Data set 2 with 640x400 pixels [3]**

Three different frame rates were considered for the two videos to test NNA and *CellREM*. The frame rate was modified by maintaining the video length (respectively 5,1 seconds for Data set 1 and 5,5 seconds for Data set 2) and removing frames between each interval. *CellREM* combination was performed starting from the NNA combination (1,0,0) and then adding weights to the other cell characteristics (shape factor and eccentricity) in the case of NNA combination didn't performed 100% of the tracking correspondence.

Table B.1 shows the comparison between the *CellREM* and NNA tracking approaches, using different frame rate values. The best *CellREM* combinations were obtained by tuning the weights of the parameters manually.

As the initial frame rate for the two datasets was higher enough, the NNA performed the tracking 100% correct in the two first comparisons. With a lower frame rate, the cells presented large displacements between frames and NNA approach performed one mismatch in each Dataset, as opposed to *CellREM* that introducing other cell characteristics still performed every track correctly.

As expected, *CellREM* never gives less accurate tracking results once in the worst scenario it assumes a NNA approach. Naturally, this process consumes more computational resources than NNA, but it also presents itself as a more complete tool for study datasets with different frame rates.

Table B.1 – Tracking accuracy results using different frame rates

<b>Dataset 1</b>			
<b>Frame rate</b>	<b>NNA</b>	<b>CellREM (combination)</b>	
10 fps	100%	100%	(1,0,0)
1,96 fps	100%	100%	(1,0,0)
0,98 fps	88%	100%	(0.7,0.2,0.1)
<b>Dataset 2</b>			
<b>Frame rate</b>	<b>NNA</b>	<b>CellREM (combination)</b>	
6 fps	100%	100%	(1,0,0)
1,09 fps	100%	100%	(1,0,0)
0,36 fps	88%	100%	(0.8,0.1,0.1)

## Conclusions

The proposed recursive error metric (CellREM) takes advantage of the reunion of some important and flexible object's parameters. If between frames, the object drastically changes one of them, the user can remove weight from that one, maintaining good tracking results. Although inappropriate frame rates can invalidate the NNA tracking approach, CellREM can still work around this problem.

Naturally, the NNA method is a particular case of the general CellREM presented in this work. This also implies that there is a higher computational cost to the CellREM

than to the simple NNA method, but the trade-off in computational speed for high accuracy in tracking is perhaps justified when other things are considered, such as the complete control of parameters. Other properties could be considered such as the pixel intensity or the boundaries, but they would interfere in the simplicity/computational cost of the algorithm.

Some natural cells' phenomena (fusion and division), along with the common image events (appearance and disappearance) may difficult the tracking process. Nevertheless, other occurrences —such as the empirical evidence that  $e$  increases in child cells just after division — may be used to further improve the CellREM algorithm, as well as an automatic tune for the best possible combination.

The principles presented herein could also be applied to other areas, such as astronomic images, since the theory remains the same. The proposed CellREM could be used both as a benchmark for comparing algorithms' segmentation capabilities or to improve algorithms' tracking by providing a quantitative value in comparison with the other objects.



## **Appendix C**

### **PGBiomed 2014 Poster Presentation**



# THE RECURSIVE ERROR CALCULATION METRIC FOR CELLULAR TRACKING

Frederico A. Carpineiro<sup>1</sup>, Pedro M. Costa<sup>1</sup>, Mario Sáenz Espinoza<sup>1,2</sup>,  
Ivo M. Silva<sup>1,2</sup> and João P. S. Cunha<sup>1,2</sup>

<sup>1</sup>Faculty of Engineering; University of Porto, Portugal | <sup>2</sup>INESC-TEC; Porto, Portugal

## BACKGROUND

- Automatic tracking of cells and their constituents is a key challenge in biomedical image processing and a need to study many important biological processes. For example, the way neural vesicles move can be correlated to some neurodegenerative conditions such as Alzheimer's, Huntington's and Parkinson diseases.<sup>[1,2]</sup>
- Manual association is an intensive and tedious labour operation that can lead to inaccurate results.<sup>[3]</sup>
- Nearest Neighbour Association (NNA) is an automatic algorithm that has low computation cost and simplicity, allowing to quickly track objects that are well individualized and do not change its position abruptly. But large displacements of objects produce poor results with NNA.<sup>[3]</sup>

## METHODOLOGY

The Recursive Error Metric (REM) was developed to upgrade and improve the NNA tracking system, considering that the object of interest keeps its morphological properties more or less constant between two consecutive frames. REM uses the following matching function (E):

$$E = \frac{\alpha D + \beta(\Delta f_c) + \gamma(\Delta e)}{\alpha + \beta + \gamma}$$

Where  $\alpha$ ,  $\beta$  and  $\gamma$  are weighting factors for distance ( $D$ ), shape factor ( $f_c$ ) and eccentricity ( $e$ ). Their complete description can be found in the following equations:

$$D = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}$$

$$\Delta f_c = \left| \frac{4\pi A_{k+1}}{P_{k+1}^2} - \frac{4\pi A_k}{P_k^2} \right|$$

$$\Delta e = \left| \frac{L_{k+1} - l_{k+1}}{L_{k+1} + l_{k+1}} - \frac{L_k - l_k}{L_k + l_k} \right|$$

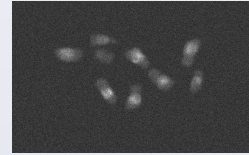
Where  $D$  is a 2D Euclidean distance (in pixels);  $f_c$  is a circularity shape factor which relates area ( $A$ ) and perimeter ( $P$ );  $e$  (eccentricity) relates the major axis ( $L$ ) with the minor axis ( $l$ ) of the objects.

The nearest neighbor association (NNA) simply uses  $D$  to associate the nearest centroid of the region of interest in the  $k$  frame to the next  $k+1$  one. Its equivalent to use the REM combination of parameters:  $\alpha = 1$ ,  $\beta = 0$  and  $\gamma = 0$ .

Some meaningful quantitative measures related to the motility, diffusivity, velocity and morphology of the moving objects can be obtained from the REM.

## PRELIMINARY CONCLUSIONS

- REM is a flexible algorithm that can be tuned to obtain the best combination of parameters to perform the tracking.
- If between frames, the object drastically changes one parameter, the user can remove weight from that one, maintaining good tracking results. Although inappropriate frame rates can invalidate the NNA tracking approach, REM can still work around this problem.
- The shape factor and the eccentricity allowed to relate several cells properties in a comprehensive, yet simple way. Other properties could be considered such as the pixel intensity or the boundaries, but they would interfere in the simplicity/computational cost of the algorithm.
- The developed methodology can be also used as a benchmark for comparing algorithms' segmentation capabilities and to improve algorithms' tracking by providing a quantitative value in the comparison.



Example of a dataset to apply the REM tracking system. At left the original image of a particular frame and at right its segmented image. As the cells are very close, some displacements between frames can cause bad results with NNA, which can be improved by considering morphological features of the cells with the REM.

## REFERENCES

- [1] T. C. Ku, Y. N. Huang, C. C. Huang, D. M. Yang, L. S. Kao, T. Y. Chiu, C. F. Hsieh, P. Y. Wu, Y. S. Tsai, and C. C. Lin, "An automated tracking system to measure the dynamic properties of vesicles in living cells", *Microscopy research and technique*, vol. 70, no. 2, pp. 119-134, 2007.
- [2] M. R. Winter, C. Fang, G. Banker, B. Roysam, and A. R. Cohen, "Axonal transport analysis using multitemporal association tracking", *International Journal of Computational Biology and Drug Design*, vol. 5, no. 1, pp. 35-48, 2012.
- [3] F. A. Carpineiro, P. M. Costa, M. Sáenz Espinoza, I. M. Silva, and J. P. S. Cunha, "NeuronDynamics: A method for neurotransmitter vesicle movement characterization in neurons", 2014, in press for IEEE International Symposium on Biomedical Imaging (ISBI).

## ACKNOWLEDGMENTS

This work was partly supported by the C-BER/BRAIN unit of INESC-TEC and is financed by the ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP - 01-0124-FEDER-037281.