

City Research Online

City St George's, University of London

Citation: Solis-Lemus, J. A. (2018). Segmentation and shape tracking of overlapping macrophages in fluorescent microscopy images. (Unpublished Doctoral thesis, City, University of London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version. To cite this item please consult the publisher's version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/22262/

Copyright and Reuse: Copyright and Moral Rights remain with the author(s) and/or copyright holders. Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge, unless otherwise indicated, provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way. For full details of reuse please refer to City Research Online policy.

City Research Online: http://openaccess.city.ac.uk/ publications@citystgeorges.ac.uk

Segmentation and Shape Tracking of Overlapping Macrophages in Fluorescent Microscopy Images



José Alonso Solís-Lemus

Supervisor: Dr Constantino Carlos Reyes-Aldasoro

School of Mathematics, Computer Science and Engineering

Research Centre for Biomedical Engineering

This dissertation is submitted for the degree of

Doctor of Philosophy

City, University of London 30th October 2018

Determination makes you run, never stop. Got to win, got to run till you drop.
Keep the pace, hold the race. The mind is getting clearer...
- Steve Harris, "The loneliness of the long distance runner"
Iron Maiden (Somewhere in Time)

Table of contents

List of figures **ix**

List of tables **xvii**

Notation & Symbols **xxv**

1 Introduction and background **1**

1.1 Overview 1
1.2 Motivation and relevance 2
1.3 Aims and objectives 3
1.4 Contributions 4
1.5 Outline and thesis structure 4
1.6 Cell migration 7
1.7 Mathematical modelling of cell motility 11
1.8 Acquisition of images of migrating cells 16
1.9 Summary 24

2 Materials **27**

2.1 Representation of images 27
2.2 Materials 31
2.3 Object boundaries and synthetic data 40
2.4 Ground truth and comparison of binary images 47
2.5 Summary 52

3 Exploration of the data **55**

3.1 Classical image analysis 56
3.2 Segmentation techniques of overlapping objects 80
3.3 Tracking 89

3.4 Summary 94

4 Junction Detection and Segmentation 97
4.1 Junctions 98
4.2 Baseline segmentation 99
4.3 The *Anglegram* Matrix 104
4.4 Junction detection with the *Anglegram* 109
4.5 Experiments with the *Anglegram* 113
4.6 Segmentation of overlapping objects 119
4.7 Summary 127

5 A Cell Shape Analysis and Tracking Algorithm: `macrosight` 129
5.1 Modification of the PhagoSight framework 131
5.2 Shape tracking 135
5.3 Analysis of the interactions of cells 138
5.4 Summary 143

6 Experiments and Results 145
6.1 ISBI Cell Tracking Challenge 146
6.2 *Artemia franciscana* dataset 152
6.3 Detection of peaks with Anglegram 155
6.4 Fluorescently labelled macrophages analysis 156
6.5 Summary 169

7 Conclusions and Future Work 175
7.1 Summary 175
7.2 Major contributions 177
7.3 Major conclusions 178
7.4 Limitations 179
7.5 Suggestions for further research 180

References 183

Appendix A Mathematical preliminaries 195
A.1 Notation 195
A.2 Vector spaces and Matrices 195
A.3 Notions of multivariate calculus 199

A.4 Set theory 200

Appendix B Preliminaries of image analysis **203**

B.1 Classical image analysis 203

B.2 Machine Learning Techniques 240

B.3 Segmentation performance metrics 242

Appendix C PhagoSight and the Keyhole tracking algorithm **245**

C.1 Segmentation. 245

C.2 Implementation of automatic thresholds for segmentation module . . . 246

C.3 Keyhole Tracking Algorithm. 247

Appendix D Introduction to PDEs in image analysis **251**

D.1 Partial differential equations in image analysis 251

D.2 Curve Evolution. 253

Index **257**

List of figures

1.1 Graphical outline of the thesis. 5
1.2 Examples of datasets analysed in this thesis. 5
1.3 Examples of the *anglegram* tool and its relationship to shapes. 6
1.4 Diagram of cellular movement. 8
1.5 Diagram of the three interrelated phenomena in chemotaxis, as depicted by Van Haaster. 10
1.6 Simplified diagram of cell migration. 12
1.7 Graphical representation of a cell modelling technique based on [33]. . . 14
1.8 Explanation of a methodology of cell modelling based on [26, 34] 14
1.9 Overview of fluorescence microscopy and the size scales visualised. . . . 17
1.10 Diagram of molecular fluorescence and setup of a confocal fluorescent microscope. 18
1.11 Images obtained by Stramer et al. [4] with two probes. 19
1.12 Examples of images obtained from rodents. 20
1.13 Artemia franciscana in traditional setup for toxicological assessment of an environment. 21
1.14 Data of neutrophils observed in zebrafish embryo. 22
1.15 *Hematopoiesis* in *Drosophila melanogaster* 23
1.16 Examples of images obtained from *D. melanogaster*. 24

2.1 Representation of the creation of a digital image. 28
2.2 Representation of RGB colour space and images represented in it. . . . 29
2.3 Example of a binary image defined from a digitised image. 30
2.4 Example frames of all the ISBI datasets in the 2015 challenge. 31
2.5 Frame taken from the C2DL-MSC dataset. 33
2.6 Frame taken from the N2DH-GOWT1 dataset. 34
2.7 Frame taken from the N2DH-SIM dataset. 34

2.8 Frame taken from the N2DL-HeLa dataset. 35
2.9 Frame taken from the C2DL-PSC dataset. 35
2.10 Layout of the *Artemia Franciscana* experiment. 36
2.11 One representative image of the *Artemia*. 37
2.12 Representative time frame of the `MACROS` datasets. 38
2.13 Mean and variance frames of the `MACROS2` dataset for the Red and Green channels. 39
2.14 Example of overlapping cells in a single frame. 39
2.15 Example of cells' shapes in a single frame. 40
2.16 Evolution of a single cell. 40
2.17 Illustration of the formation of a clump using ellipses. 41
2.18 Examples of ellipse boundaries generated by equation (2.2). 43
2.19 An example of the generation of overlapping binary ellipses. 44
2.20 Overview of the range of paired ellipses investigated. 44
2.21 Synthetic generation of random basic shapes using splines. 46
2.22 Synthetic generation of random basic shapes. 46
2.23 *Pointiness* changes observed in consecutive frames. 47
2.24 Diagram of the manual annotating software and types of annotated data 48
2.25 Example of the ground truth at a representative time frame. 49
2.26 Graphical representation of the Jaccard index, Precision and Recall measurements. 50
2.27 Procedure to compare ground truth (GT) of overlapping macrophages with segmentation output. 51

3.1 Presentation of **test frame** and the cropped images used in testing throughout this chapter. 56
3.2 Overview of the preprocessing methodologies analysed in this work. . . 57
3.3 Comparison of histogram modification techniques with detail of the green channel of a macrophage image. 58
3.4 Examples of size reduction using both things proposed. 59
3.5 Examples of transformations on detail of a frame. 59
3.6 Simplified diagram of a 2D convolution of an image and a 5×5 kernel. 60
3.7 Representation of filters for the Gaussian kernel with different variance σ^2. 62
3.8 Representation of order different statistic filters with varying sizes. . . . 63
3.9 Relationship between derivatives and filters. 64

3.10 Representation of different sharpening filters. 65
3.11 Representation of the different implementations of the Laplacian filter for $\mathbb{K}_1$ and $\mathbb{K}_2$. 66
3.12 Example of the minimum eigenvalue of the Hessian matrix per pixel. . 66
3.13 Detail of output of Canny algorithm on macrophages data. 67
3.14 Detail of two macrophages after applying different Otsu thresholds. . . 69
3.15 Limitations of the Otsu algorithm in frame from the ISBI dataset. . . . 69
3.16 Test frame after applying different adaptive threshold techniques. . . . 70
3.17 Representation of different structural elements in abstract terms and as a binary kernel $\mathbb{S}$. 71
3.18 Diagram of morphological operations. 72
3.19 Example of morphological operations outputs when applying different structural elements. 73
3.20 Example of a Gabor filter bank for feature extraction. 75
3.21 Convolution output of the different Gabor kernels on the green channel. 75
3.22 Foreground and background detected using Gabor features. 76
3.23 Cell in two consecutive frames for experimentation of SURF in `MACROS1` dataset. 78
3.24 Experimentation of SURF algorithm in macrophages data using different parameters of threshold and number of scales. 79
3.25 Experimentation of SURF algorithm from the region surrounding the reference cell to the target frame. 79
3.26 Illustration of a Voronoi Partition. 81
3.27 Naïve segmentation using Voronoi Tesselations. 82
3.28 Results from implementation of Voronoi partition in overlapping macrophages. 82
3.29 Graphical representation of the ASM algorithm. 83
3.30 Landmark points determination in different instances of macrophages. . 84
3.31 Implementation of the ASM methodology in macrophages data. 85
3.32 Results of the ASM model implementation. 86
3.33 Overview of the framework proposed by Lu et al. [99]. 87
3.34 Description of input presented to joint level set method. 88
3.35 Comparison of the joint Level Set Method with the Ground Truth and a Voronoi Partition. 89
3.36 Illustration of tracking experiments. 90
3.37 Example of implementation of PhagoSight on different ISBI data. . . . 92

3.38 Preprocessing stage on the *Artemia Franciscana* dataset. 93
3.39 Implementation of PhagoSight on the *Artemia* data. 93
3.40 Implementation of PhagoSight on Red channel or `MACROS1` data. 94

4.1 Diagram representing the two distinct types of junctions recognised in the working datasets. 98
4.2 Diagram of baseline segmentation. 100
4.3 Comparison of the Jaccard index for each object detected, whether a clump or not, at each of the ten consecutive frames with available GT. 101
4.4 Detection of objects in the green channel on three representative frames of the `MACROS1` dataset. 102
4.5 Representation of possible partition of a clump based on the position of its junctions. 103
4.6 Illustration of the arrangement of points in a boundary and the calculation of the inner angles of two points. 105
4.7 Graphical representation of the calculation of the inner point angle of point $\mathbf{p}_i$ at separation j. 106
4.8 Explanation of inner angle of a point in the construction of the anglegram. 107
4.9 Representation of *inner point angle* calculation and generation of *anglegram matrix*. 108
4.10 Graphical representation of the detection of junctions via the *anglegram*. 109
4.11 Tests from the different row projections. 110
4.12 Junction detection on overlapping objects through the maximum intensity projection of the anglegram matrix. 111
4.13 Anglegram of basic shapes. 112
4.14 Minimum intensity projection of the anglegram of basic shapes. 112
4.15 Results of the corner detection through the *anglegram* method for overlapping ellipses. 114
4.16 Cases where the detection of corners was not accurate. 115
4.17 Results of the corner detection through the *anglegram* method for clumps in macrophages data. 115
4.18 A random sample of the synthetic data used to test the peak detection using *anglegram*. 116
4.19 *Pointiness* changes observed in consecutive frames. 117
4.20 Assessment of *pointiness* of the basic shapes. 118

4.21 Qualitative comparison of junction detection via anglegram versus the Harris corner detector. . . . 119
4.22 Initial study comparing Jaccard indexes of the segmentation via Edge Following versus Voronoi tessellations. . . . 121
4.23 Initial study comparing segmentation via Edge Following and Voronoi tessellations. . . . 122
4.24 Illustration of all the methods developed and the workflow to obtain results. . . . 124
4.25 Qualitative comparison of different segmentation methods in two different clumps. . . . 125
4.26 Comparison of Precision, Recall and Jaccard index for all segmentation methods presented in this section. . . . 127

5.1 Complete graphical abstract of `macrosight`. . . . 130
5.2 Illustration of clump codes incorporated to a particular track information. 134
5.3 Illustration of shape tracking functionality. . . . 135
5.4 Comparison between the shrink and grow states of the **evolve** function. 137
5.5 Outline of the main hypothesis in this work. . . . 139
5.6 Graphical illustration of the analysis done in this work. . . . 139
5.7 Diagram explaining the calculation of the θ_x angle. . . . 140
5.8 Diagram explaining the time frames chosen from tracks $\mathcal{T}_r, \mathcal{T}_q$ for the analysis of direction change. . . . 141
5.9 Measurement of angle change with cell-cell contact (i) and control (ii) experiments. . . . 142

6.1 Example frames of all the ISBI datasets in the 2015 challenge. . . . 147
6.2 Preprocessing by shading correction applied to phase contrast dataset. . 149
6.3 Comparison between shading correction and adaptive thresholding. . . 149
6.4 Segmentation results submitted to the ISBI Cell Tracking Challenge. . 150
6.5 Tracking results submitted to the ISBI Cell Tracking Challenge. . . . 151
6.6 Postprocessing of the tracks resulting from inconsistencies in the segmentation. . . . 153
6.7 Detail of the tracks in the *Artemia franciscana* dataset. . . . 153
6.8 Illustration of the visualisation of measurements. . . . 154
6.9 Complete tracks obtained from red channel in all three datasets. . . . 157

6.10 Overview of the three experiments presented in this section regarding the shape tracking analysis. . . . 158
6.11 Evolution of cell shapes throughout multiple frames. . . . 159
6.12 Comparison of the direction changes of tracks with or without a *clump* interaction. . . . 161
6.13 Change of direction before and after a clump. . . . 163
6.14 Comparison of the time in clump and the angle of variation. . . . 164
6.15 Comparison of angle change, depending on the time spent in clumps. . 165
6.16 Representation of two cells in clump `2001` through the direction change and interaction analysis. . . . 166
6.17 Representation of two cells in clump `3002` through the direction change and interaction analysis. . . . 167
6.18 Representation of two cells in clump `22001` through the direction change and interaction analysis. . . . 168
6.19 Frames in clump interaction overlapped to appreciate cell movement. . 169

A.1 Detailed diagram of convolution in 2D variables. . . . 198
A.2 Discretisation of a central point derivative in 1D variables. . . . 199
A.3 Representation of the mean and variance frames in the Artemia Franciscana dataset. . . . 202
A.4 Representation of a histogram of the intensity values of an image. . . . 202

B.1 Overview of the preprocessing methodologies analysed in this work. . . 204
B.2 Illustration of size reduction in an image. . . . 206
B.3 Representation of the process which generates a Gaussian pyramid in one dimension. . . . 206
B.4 Examples of size reduction using both things proposed. . . . 206
B.5 Examples of transformations on detail of a frame. . . . 208
B.6 Simplified diagram of a 2D convolution of an image and a 5×5 kernel. 209
B.7 Representation of order different statistic filters with varying sizes. . . . 211
B.8 Test frame after applying smoothing through different Order statistic filters. . . . 212
B.9 Relationship between derivatives and filters. . . . 213
B.10 Explanation of the image gradient. . . . 213
B.11 Representation of different sharpening filters. . . . 216
B.12 Test frame after applying smoothing through different filters. . . . 216

B.13 Representation of the different implementations of the Laplacian filter for $\mathbb{K}_1$ and $\mathbb{K}_2$. 218
B.14 Example of the minimum eigenvalue of the Hessian matrix per pixel. . 218
B.15 Detail of output of Canny algorithm on macrophages data. 220
B.16 Diagram of hysteresis in one dimension. 222
B.17 Detail of hysteresis threshold in macrophages frame. 222
B.18 Illustration of a hysteresis threshold. 223
B.19 Detail of two macrophages after applying different Otsu thresholds. . . 224
B.20 Limitations of the Otsu algorithm in frame from the ISBI dataset. . . . 225
B.21 Graphical explanation of the calculation of parameter $\mu_s^{x_s}$. 227
B.22 Representation of a structural element in abstract terms and as a binary kernel $\mathbb{S}$. 229
B.23 Examples of structural elements. 229
B.24 Diagram of erosion and dilation operations. 231
B.25 Example of erosion and dilation outputs based on the different structural elements shown in B.23. 231
B.26 Diagram of morphological opening and closing operations. 232
B.27 Examples of opening and closing performed using different structural elements on B.23. 233
B.28 Example of a Gabor filter bank for feature extraction. 234
B.29 Convolution output of the different Gabor kernels on the green channel. 235
B.30 Foreground and background detected using Gabor features. 235
B.31 Setup of generic SURF implementation. 238
B.32 Illustration of SURF features detection and matching. 239
B.33 Common network topologies used for SOMs. 240
B.34 Layout of the CNN architecture. 242
B.35 Diagram of comparison of ground truth against segmentation output. . 243
B.36 Graphical representation of the Jaccard index, Precision and Recall measurements. 244

C.1 PhagoSight user interface for selecting manual thresholds. 246
C.2 Illustration of the Keyhole algorithm 248

D.1 Tangential and normal components to the velocity of a curve. 253
D.2 Representation of two iconic curve evolution models. 254

List of tables

2.1 Description of the ISBI datasets used for this work. 32
2.2 Values assigned to parameters of the ellipses to generate binary images. 44

3.1 Histogram modification functions. 58
3.2 Table of kernels of common sharpening filters in different directions. . . 65
3.3 Description of main morphological operations. 72
3.4 Track information retrieved from PhagoSight framework. 91
3.5 Main conclusions from the data exploration with traditional image segmentation. 95

4.1 Synthetic test of corner detection from 139 cases of overlapping 114
4.2 Synthetic test of peak detection. 117
4.3 Comparison of mean values of Precision, Recall and Jaccard Index for clumps 2 and 3 over 10 frames. 126
4.4 Statistical analysis of results presented in table 4.3. 126

5.1 Track information retrieved from PhagoSight framework. 133
5.2 Brief description of the parameters measured per track. 133
5.3 Examples of clump codes created through the track labels obtained by PhagoSight by applying the defined codes. 134
5.4 Parameters used of the active contour function based on the desired state required. 137

6.1 Comparison of the ISBI Challenge results when changing the method of defining the hysteresis thresholds. 148
6.2 Peak detection using the Harris algorithm. 155
6.3 Comparison of *anglegram* vs. Harris corner detector. 156
6.4 Angle change (θ_x) comparison per dataset. 162

List of tables

A.1 Basic set operations 200
A.2 Basic set operations 201

B.1 Histogram modification functions. 205
B.2 Geometric transformations represented in 3×3 matrices. 207
B.3 Choice of parameters for the generation of a scale space. 237

C.1 Track information retrieved from PhagoSight framework. 249

D.1 Different notations used to represent partial differential equations. . . . 251

Acknowledgements

This work was conducted within the Research Centre for Biomedical Engineering (RCBE) in the School of Mathematics, Computer Science and Engineering at City, University of London (City) with the support of a PhD Studentship provided by *City, University of London*.

I am very grateful to my supervisor, Dr Constantino Carlos Reyes-Aldasoro, for the trust given when selecting me to receive the PhD Studentship. I am also grateful for his constant guidance and support throughout my studies at City, going above and beyond for to assist me in becoming an integral academic and researcher. I am honoured to have been his student and aspire to be a scientist like him.

I would like to thank Dr Greg Slabaugh, my former second supervisor, who always provided keen insights and precise advise in the various stages of my work. I would also like to thank Dr Brian Stramer from *Randall Centre for Cell & Molecular Biophysics* at *Kings College London* for providing the excitingly challenging datasets of fluorescently labelled macrophages. I would like to acknowledge Professor Kyriacou and all the members, academic and researchers, of RCBE.

I worked with many people throughout my years working at RCBE, notably: Zaibaa, Tomas, Mario, James, Michelle H, Loukas, John, Subhasri, Zaheer and Cefa. I enjoyed sharing these years and will always open the door of the lab for you.

Finally and most importantly, I dedicate this thesis and all my accomplishments to my parents. To my mother for giving me all her life, unrestricted and unconditional, to my father for teaching me hard work and dedication. All my abilities, talents or achievements, whether in the past, present or future are built on their hard effort. I also want to thank my sister for her company, advise and being an inspiring role model. Finally, I want to thank Zaibaa simply for all and for everything, you are my best friend, my *ghost*.

Declaration

I hereby declare that the work presented in this thesis is my own work. Any idea, result, or illustration originating from other subject's work has been acknowledged in the text by referencing to the original author. This thesis has never been published or submitted elsewhere for obtaining an academic degree or professional qualification. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

I grant power of discretion to the Librarian at City, University of London to allow the thesis to be copied in whole or in part without further reference. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

José Alonso Solís-Lemus
30th October 2018

Abstract

This thesis investigates the complex movement of migrating cells and small organisms. Whilst special emphasis has been devoted to macrophages observed with fluorescent confocal microscopy, the segmentation, tracking and shape analysis techniques developed here can be extended to various other applications, not exclusively of cells. With the acquisition conditions and particular resolution, macrophages exhibit very complex shapes. In some cases, the cells overlap, complicating the analysis considerably.

A software framework is proposed to detect, track and analyse the shape and movement of macrophages. The framework, called `macrosight`, includes the following: (i) a novel 2D matrix with multiscale angle variation, called the *anglegram*, based on the angles between the points of the boundary of an object through which its general shape can be characterised. For the case of the macrophages, the *anglegram* helped identify the intersection of overlapping cell boundaries. (ii) A cell shape tracking algorithm providing a measured evolution of shape parameters and categorisation of shape types. This was based on metrics extracted from the *anglegram* and used for non-overlapping cells. (iii) An algorithm to assess the changes of direction derived from cell-cell contact. The algorithm may assist in the identification of cell-cell interactions, which may influence migration patterns.

The *anglegram* showed promising results identifying two types of junctions in the boundaries of segmented objects, both in synthetic and real data. Tests on synthetic data showed that the method outperformed the classic junction detection technique by Harris. The framework can be used to explore the influence of cell-cell contact in movement patterns. The early experiments presented provided statistically significant evidence of such influence, opening the path for additional information of the tracks being included to further investigate on a larger number of interactions.

This thesis provides preliminary studies of macrophages' shapes and interactions. Future developments incorporating the techniques proposed in this thesis would provide a precise analysis of cell movement in biological studies, which could lead to pioneering insights concerning cell migration.

Notation & Symbols

Mathematical Notation

$\circledast$	Convolution operator.
$\|\cdot\|_2$	Euclidean norm.
∂_x, ∂_y	Partial derivatives
$\langle \cdot, \cdot \rangle$	Scalar product operator.
$c(s)$	Parametric curve with parameter s.
$\mathcal{B}$	Set of points describing the boundary of an object.
x, y	Discrete variables used to address positions in an image.
N_c	Number of columns in an image.
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution (Gaussian) with a mean μ and variance σ^2.
s, p	Parameters in 2D curves.
$\mathbb{R}$	Set of real numbers.
$R(\theta)$	Rotation matrix, rotating by angle θ.
N_r	Number of rows in an image.
t	Referring to time, iterations or time frame in image sequences.
$\mathbb{I}$	Matrix representing an image.
$\mathbb{K}, \mathbb{S}$	Matrices representing kernels or 2D filters.
px, qx	Representing the value of at image at a specific position.

$\mathbf{x}, \mathbf{v}, \mathbf{p}$	Vectors
FN	Number of false negatives.
FP	Number of false positives.
TN	Number of true negatives.
TP	Number of true positives.

Matlab functions

findpeaks	Find local maxima in 1D function.
imfilt$(\mathbb{I}, \mathbb{K})$	Filtering of image $\mathbb{I}$ with kernel $\mathbb{K}$.
imfreehand	Crreates a free hand region in an image.
imgradient	Calculation of the image gradient.
imresize	Resizes an image to a given scale.
mean	Representation of the mean function.
poly2mask	Representing a boundary into a binary image.
regionprops	Region properties.
std	Representation of the standard deviation function.

Notation specific to `macrosight`

$\theta_{i,j}$	Inner angle of point $\mathbf{x}_i$ at separation j.
Θ	Anglegram matrix
$\hat{\theta}_{Pr}$	Projection of Θ along its rows.
$c(r, q)$	Identifier for clump containing tracks $\mathcal{T}_r$ and $\mathcal{T}_q$.
DS	Dataset analysed by `macrosight` framework.
DS_{Re}	Dataset resulting of segmenting and labelling images of DS_{Re}.
DS_{Re}	Dataset resulting of preprocessing images of DS.

$\mathcal{T}$	Track identifier.
$\bar{\mathcal{B}}$	Representation of translated boundary for shape evolution algorithm.
θ_x	Change of direction angle used in interaction analysis.
S	Number of frames before and after a clump, selected for interaction analysis.

Acronyms / Abbreviations

EF	Edge Following (in overlapping segmentation via anglegram)
JS	Junction Slicing (in overlapping segmentation via anglegram)
SOM fitting	Self-Organising Map fitting (in overlapping segmentation via anglegram)
ASM	Active Shape Models
BMVC	British Machine Vision Conference
CNN	Convolutional Neural Network
CT Challenge	Cell Tracking Challenge
DIC	Differential interference microscopy
DoG	Difference of Gaussians (SIFT)
Fluo	Fluorescence microscopy
fps	Frames-per-second
GFP	Green Fluorescent Protein
GT	Ground Truth
GUI	Graphical User Interface
IEEE	Institute of Electrical and Electronic Engineers
ISBI	IEEE Symposium on Biomedical Imaging

JI	Jaccard similarity Index
`MACROS`	Referring to a macrophages dataset.
MATLAB®	Numerical computing environment and programming language developed by MathWorks
MVAB	Machine Vision of Animals and their Behaviour
PCA	Principal Component Analysis
PhC	Phase contrast microscopy
PhS	Abbreviation for the PhagoSight framework
RGB	Red, Green and Blue images
ROI	Region of Interest
SIFT	Scale-Invariant Feature Transform
SNR	Signal to Noise Ratio
SOM	Self-Organising Maps
SURF	Speeded-Up Robust Features
MIP	Maximum Intensity Projection
mIP	Minimum Intensity Projection
Δ direction	Change of direction
Post-clump	Referring to collection frames after two cells interact in a clump.
Pre-clump	Referring to collection frames before two cells interact in a clump.

List of publications

Journal publications:

- C Mitchell, L Caroff, A Vigilante, **JA Solis-Lemus**, CC Reyes-Aldasoro, F de Chaumont, A Dufour, S Delongeville, JC Olivo-Marin, R Knight, 'Cell Tracking Profiler: a user-driven imaging analysis framework for statistical analysis of cell behaviour in 3D time-lapse datasets', *Nature Communications*, (Submitted)
- **JA Solís-Lemus**, B Stramer, G Slabaugh, and CC Reyes-Aldasoro, 'Macrosight: A Novel Framework to Analyse the Shape and Movement of Interacting Macrophages', *Journal of Imaging*, vol. 5, no. 17, 2019.
- Ulman, Maška, Magnusson, Ronneberger, Haubold, Harder, Matula, Matula, Svoboda, Radojevic, Smal, Rohr, Jaldén, Blau, Dzyubachyk, Lelieveldt, Xiao, Li, Cho, Dufour, Olivo-Marin, Reyes-Aldasoro, **Solís-Lemus**, $\cdots$ and Ortiz-de-Solorzano, 'An objective comparison of cell-tracking algorithms', *Nature Methods*, vol. 14, p. 1141, Oct. 2017.
- **JA Solís-Lemus**, B Stramer, G Slabaugh, and CC Reyes-Aldasoro, 'Segmentation and Shape Analysis of Macrophages Using Anglegram Analysis', *Journal of Imaging*, vol. 4, no. 1, p. 2, Dec. 2017.

Conference publications:

- **JA Solís-Lemus**, B Stramer, G Slabaugh, and CC Reyes-Aldasoro, 'Analysis of the Interactions of Migrating Macrophages', in *Medical Image Understanding and Analysis (MIUA 2018)*, 2018, pp. 262–273.
- **JA Solís-Lemus**, B Stramer, G Slabaugh, and CC Reyes-Aldasoro, 'Shape analysis and tracking of migrating macrophages', in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 1006–1009.

- **JA Solís-Lemus**, B Stramer, G Slabaugh, and CC Reyes-Aldasoro, 'Segmentation of Overlapping Macrophages Using Anglegram Analysis', in *Medical Image Understanding and Analysis (MIUA 2017)*, 2017, pp. 792–803.
- **JA Solís-Lemus**, Y Huang, D Wlodkowic, and CC Reyes-Aldasoro, 'Microfluidic environment and tracking analysis for the observation of Artemia Franciscana', in *Proceedings of the Machine Vision of Animals and their Behaviour (MVAB)*, 2015, pp. 4.1–4.8.

Chapter 1

Introduction and background

1.1 Overview

Biomedical image analysis is a branch of Biomedical Engineering where computational algorithms are developed to analyse images (which can include image sequences, volumes or volume sequences) of different biological nature and context. Observation of biological processes through microscopy has allowed for "large-scale, systematic and automated studies", as stated by Rittscher et al. [1]. For example, experiments where the observation of cellular processes in genetically modified organisms produces several images per second. Manual assessment of such experiments is a difficult and time consuming task, for which an automated or semi-automated technique could obtain useful information from the data. To extract information from an image, a collection of mathematical and statistical procedures can be performed on the images acquired, and the information obtained can aid tasks such as cell detection, counting and tracking.

The migration of cells is of great importance in many biological processes, one of them is within the immune system [2, 3]. Under normal conditions, cells can migrate in the body in response to a directional cue. The nature of these cues categorise motion in terms of its nature, such as chemotactic (induced by chemical substances known as chemoattractants), electrotactic (induced by electric fields) or durotactic (due to differences in pliability). This work will focus on movement through chemotaxis, which is involved in the immune response of the system in wound repair processes.Injured cells secrete small proteins called *cytokines*, which act as pro-inflammatory signals (*chemokines*) triggering the movement of immune cells from the lymph nodes to the circulatory system and from there to the site of injury. Macrophages are one of the

cells within the immune system that settle in lymphoid tissues and the liver, serving as filters for trapping microbes and foreign particles [2].

The experiments to acquire the video sequences are usually performed in surrogate animals, called *model organisms*. One of such animals is the common fruit fly, *Drosophila melanogaster*, allowing for several applications including the observation of macrophages. As stated by Stramer et al. [4], "Drosophila melanogaster macrophages are highly migratory cells that lend themselves beautifully to high resolution *in vivo* imaging experiments" where the dynamic interplay of cytoskeletal structures can be observed as the macrophages migrate.

The work documented in this thesis contributes towards (i) understanding movement patterns of macrophages and other cells, (ii) providing measurements of the movement in relation to the shape of the cells and (iii) automating the task through computer algorithms. The thesis investigate image segmentation and tracking of time sequences of fluorescently labelled macrophages and provides analysis of the movement, in the context of their changing shapes and interactions. The novel computational algorithms described in this thesis extract information from the video sequences and provide statistical analysis of its results.

1.2 Motivation and relevance

Cell migration is an essential biological process that ensures homeostasis in adults, where an unbalanced migratory response results in human disease; with excessive migration causing autoimmune diseases and cancer metastasis [5]. As shown by Petrie et al. [6], although the basic mechanisms of random cell movement are well characterised, no single model explains the complex regulation of directional migration. A better understanding of the processes that drive the movement of cells could lead to improved strategies for treatment.

Stramer et al. [4] acquired images of macrophages in embryos of the common fruit fly, which were genetically modified to produce two colours of fluorescence: nuclei appearing in red and the microtubules fluorescing in green. Thus showing through *in vivo* experimentation and observation of the green channel that macrophages form an array of cytoskeletal structures which appear to indicate the direction of migration. The arm-like arrays in two cells appear to align just before cell-cell repulsion, indicating that a component in the moving patterns of cells might not be taken into consideration, *i.e.* the interactions of the cells appear to anticipate the direction of migration. A

quantitative analysis of such phenomena could provide statistical proof of the migration patterns of cells related to shape changes and interactions.

Acquiring time sequences of moving cells is a complex task in itself, and it presents challenges for automated image analysis algorithms. Various sources of noise can be identified at different stages of the acquisition. Some examples include (i) the digital camera not producing high quality images; (ii) out of focus samples, causing microscopes to blur some areas of the image or undesired shading; or (iii) the staining of the desired cellular structures not being uniform, or in case of fluorescence, the intensities could be varied.

Problem statement. The cells in the images obtained by Stramer et al. [4] present additional challenges for automated analysis that is not necessarily shared with other tracking problems widely studied in computer vision. For example, (i) the cell membrane is not present in the images and thus the overall shape of the cells tend to be difficult to observe. (ii) Some events of cell-cell contact are of great interest, however these cause cells to overlap causing an ambiguity in the boundaries of the cell bodies. The human vision can often overcome some of these challenges, however, a computational analysis would require complex methods to analyse each frame. The techniques developed for this purpose would provide robust and repeatable analysis to image sequences of migrating macrophages and other cells, aiding in the confirmation of the hypotheses made from visual inspection.

1.3 Aims and objectives

The aim of this thesis is to develop a software framework that will allow a more specialised analysis of the movement of migrating macrophages obtained from image sequences, solving the ambiguities introduced in the imaging. The following objectives are presented:

1. Develop a software framework for the tracking of shapes of fluorescently labelled macrophages as they evolve in time. This framework will involve the detection of the foreground, tracking the simpler red channel, and using the resulting information to track the shape and analyse the movement of the corresponding cells in the green channel.

2. Describe, within the proposed framework, experiments to test whether cell-cell contact events in the available datasets correspond to specific cell interactions influencing migration.

3. Describe an accurate geometry-based image analysis methodology to analyse the shapes of the macrophages and classify them into different states, which could be correlated to the temporal context (tracks) and the spatial context (relating shape states to proximity to other cells).

4. Propose a method to disambiguate images of macrophages that overlap, as this would allow for the analysis of movement and shape to be applied when the cells appear to be interacting with each other.

The objectives presented are oriented towards building a solid research foundation in which questions about the nature of the movement can be raised.

1.4 Contributions

The original contributions of this thesis are:

1. A thorough exploration of the macrophages data through classic and novel image analysis techniques.

2. Software tool for the analysis of shape, which detects peaks and corners in the shapes of detected cells, or *clumps* of cells. The software tool could work as a signature of a particular shape.

3. Development and testing of a software framework specific for, but not limited to, the analysis of migrating macrophages, distributed as a set of routines under an open source license. Such framework would directly address the aims and objectives described in the previous section.

1.5 Outline and thesis structure

An overview of the thesis is represented graphically in Figure 1.1, and subdivided in the chapters described below.

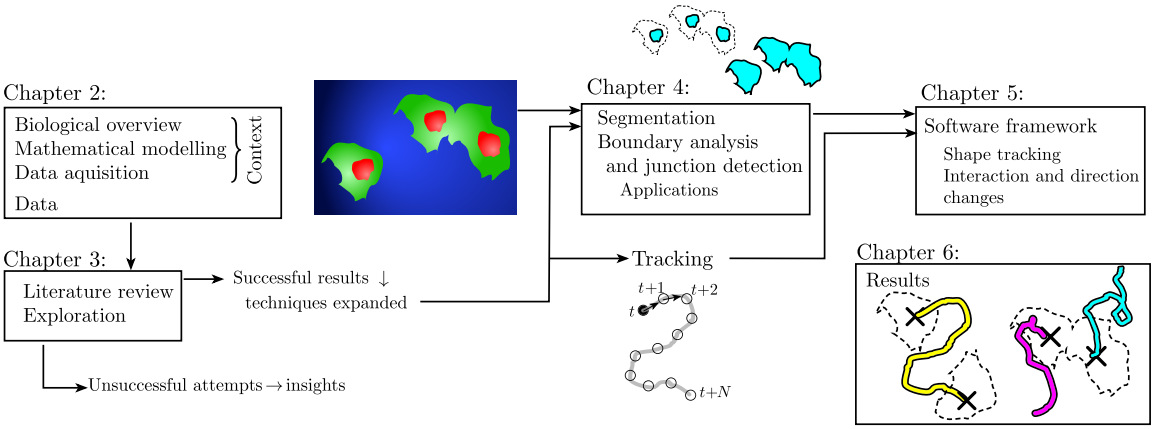
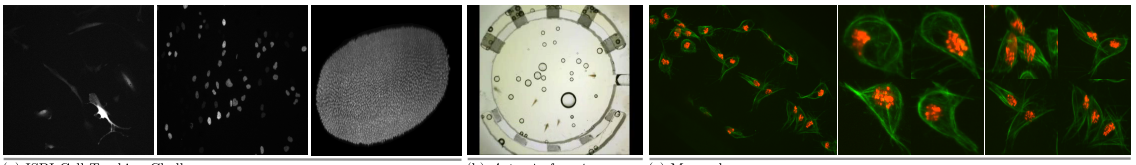


Figure 1.1 Graphical outline of the thesis. Chapter 2 provides the context for cell migration from a biological and mathematical point of view. Chapter 3 represents a thorough exploration of the data. Chapters 4 and 5 provide the major contributions of this work, the results are presented in Chapter 6.

Chapter 1. The second half of this chapter presents background knowledge, in which **cell migration** as a complex biological process is presented alongside the mathematical models used to describe it. The acquisition of images and time sequences through confocal fluorescence microscopy is also discussed.

Chapter 2. The data is presented, consisting of three datasets containing time sequences of different types: (i) the Cell Tracking Challenge of the IEEE Symposium of Biomedical Imaging (ISBI) in 2015, (ii) images of brine shrimp, *Artemia franciscana*, small marine crustaceans utilised in toxicological experiments, (iii) the fluorescently labelled macrophages datasets. The generation of synthetic data and acquisition of ground truth (GT) for images of macrophages is introduced.



(a) ISBI Cell Tracking Challenge (b) *Artemia franciscana* (c) Macrophages

Figure 1.2 Examples of datasets analysed in this thesis. (a) ISBI Cell tracking challenge data, 3 examples out of 15 datasets shown. (b) *Artemia franciscana* in a toxicological experiment. (c) Macrophages: full frame and detail of cell-cell interactions and shape changes.

Chapter 3. A thorough exploration of the datasets described in Chapter 2, especially of the macrophages data, using classic and novel image analysis techniques. The aim

of the chapter is to present a literature review of image analysis techniques alongside an exploration of the data, gaining insights for the macrophages dataset.

Chapter 4. A novel tool for the analysis of the boundaries of shapes is presented, called the *anglegram*. Its main functionality derives from the need to acquire junctions in the boundaries of overlapping 2D objects. Applications for the proposed tool are presented for segmenting overlapping objects and to analyse the shape of single cells. Figure 1.3 shows examples of the anglegram representing different shape types and one case of the boundary of an overlapping object.

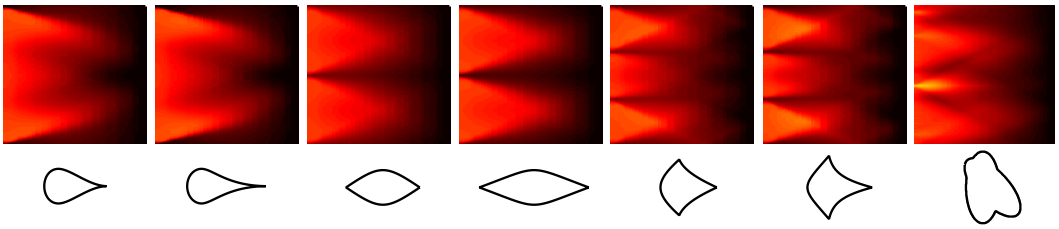


Figure 1.3 Examples of the *anglegram* tool and its relationship to shapes.

Chapter 5. The software framework `macrosight` is presented in this chapter. The chapter is subdivided in two projects: (i) analysis and tracking of the shape evolution of single cells, recognising four classes of shapes the cells can adopt; and (ii) analysis of the change of direction of cells before and after cell-to-cell contact.

Chapter 6. The results for all experiments performed on all datasets are presented here. First, the data presented in Chapter 2 corresponding to the (i) ISBI Cell Tracking Challenge and (ii) the toxicological study in *Artemia franciscana*. Finally, the data of migrating macrophages is analysed with the `macrosight` framework.

The following sections present an overview of cell migration, with some examples of moving cells from previous work done. Special emphasis is provided for cells of the immune system, neutrophils and macrophages. Cellular motion and its links to biological processes are discussed; as a comparison, mathematical modelling of cell migration is overviewed. The process of image acquisition of moving cells is described, providing an overview of model organisms, describing various imaging techniques, and how a microscope produces the images, which are the main data analysed in this work. To finalise the background overview of this thesis, the process of image acquisition of interacting macrophages through model organisms is outlined. The interactions observed raise the problems of geometry-based image analysis and tracking.

1.6 Cell migration

Cells are the smallest functional unit of an organism, which are typically microscopic and can be categorised in two groups, eukaryotic cells, containing a nucleus and prokaryotic cells which do not [7]. In this work, eukaryotes will be referred to organisms which are multicellular, and whose different cells are classified based on their specific functions.

Cell migration is an important process that involves the translation of a cell from one place to another; occurring in various environments and circumstances and due to multiple situations. As mentioned by Petrie et al. [6], the basics of cell motility are well understood, however, it is less clear how they are coupled into a procedure that integrates both external (environmental) cues and characteristics of the cells into a directionally persistent migration. In simple terms, the movement pattern observed in some migrating cells appears to be oriented, yet the mechanisms for movement are not sufficient to explain such *behaviour*.

Cell migration occurs as a response to two sources: internal and external. Internal sources correspond to a cell responding to a non-directional signal that triggers the motility processes of the cell. External sources correspond to an environmental factor, which might trigger the same processes. The nature of cues that provoke this movement categorises further the motion. Such categories are presented alongside an overall description of the process in section 1.6.

The process of cellular movement

The process in which a cell moves from one place to another is described in this section. Cells generally move on the **substrate**, which is the surface where the cells live and are supported by extra-cellular molecules, called the **extra-cellular matrix**; the observed direction the cell is moving is referred as the **gradient**. Notice that in this work, the concept of a gradient is referred to by its mathematical connotation, and only in this chapter is it mentioned in biological context. The previous concepts were briefly described, as they will be referred to in this section. A deeper analysis is not relevant to the scope of this work.

Specific molecular mechanisms operate at each step of cell motility, *i.e.* polarisation, protrusion and adhesion, translocation of the cell body and retraction of the rear that control cell migration [6]. Polarisation refers to the spatial arrangement and composition of these domains that facilitate cellular processes, including cell migration [8]. Protrusion and adhesion refer to the two steps in which forward movement of cells

are classified [9]. Finally, the translocation of the cell body and retraction of the rear can be explained as the rearranging of the cell organelles as the cell moves forward, finalising with the movement of the rear of the cell relative to the movement vector. Figure 1.4 shows an overview of the generalised process of cellular movement, based on the works by Lauffenburger and Horwitz [10] and Ridley et al. [11].

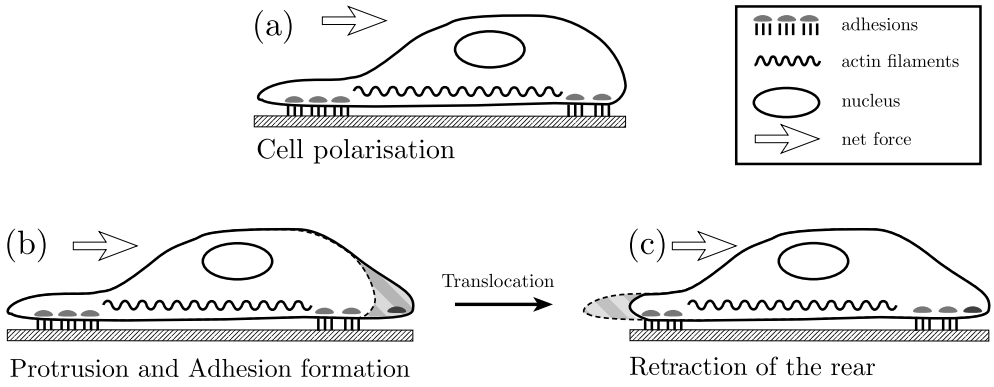


Figure 1.4 Diagram of cellular movement. (a) Polarity is intrinsic to a migrating cell [11] which results in reorganisation of organelles as well as the microtubules. (b) Migration starts with the formation of a protrusion, which is stabilised by the formation of adhesions. At the rear (c), the cell retracts as the adhesions disassemble. This diagram was based on [10, 11]

In particular, polarisation is a component process of cell migration of major interest [12]. In simple terms, polarisation involves the process in which a distinct and stable front and rear of the cell can be identified [13]. The leading edge can be identified by the intense actin polymerisation, creating a protrusive structure in the cell, and adhesion to the substratum. The trailing edge, on the other hand, is identified by the stable bundles and the release and disassembly of adhesions. Migration often arises from environments that provide a directional cue; these can be varied and are classified in five groups: haptotaxis, mechanotaxis, electrotaxis, durotaxis and chemotaxis, which are described in the following paragraphs. In particular, chemotaxis is of high relevance to this work and will be described in more detail, as it is the guiding cue for macrophages.

Haptotaxis. The process of haptotaxis occurs in response to the graded adhesion in the underlying substrate [6]. The gradient in the substrate that cause the process, can be understood as the differences in the medium in which the cell moves (substrate) causing a different reaction from the cell's *touch* with the substrate at each point. In other cases, haptotaxis occurs when other guidance cues are anchored within the

extra-cellular matrix (ECM) [6]. Such guidance cues can involve chemicals which are bound to the substrate, and causing the cell to polarise.

Mechanotaxis. The process of mechanotaxis is the result of mechanical forces applied to the substrate. For example, in epithelial cells, lining the outer surfaces of organs, fluid shear stress can affect migration patterns. Also, in wound healing processes, breakdown of cell-cell contact is a form of mechanotaxis.

Durotaxis is a process in which the changes of pliability or rigidity of the substrate cause the polarisation. Durotaxis is considered a subset of mechanotaxis [14]. The process can be described as a migrating cell having a *preference* over stiff substrates, to the point where, as described by Lo et al. [14], migration of cells can be guided by manipulating the flexible substrate.

Electrotaxis. Electrotaxis is the polarisation process in which the cell responds to electric fields. Injury that disrupts an epithelial layer instantaneously generates endogenous electric fields [15]. As Zhao [15] describes using experimental models with other directional cues (e.g., contact inhibition release, population pressure etc.) are present, electric fields of physiological strength override them and direct cell migration.

Chemotaxis. Chemotaxis involves the movement in response to chemical cues. Given the presence of extracellular signals, cells guide their movement through structures which form at the leading edge in response to the local accumulation of certain signalling molecules [16]. Chemotaxis occurs when a soluble factor is applied asymmetrically and dictates the direction of cell migration [6]. In chemotaxis, three phenomena can be observed. (a) **Self-organising pseudopodia**, which involves the movement and extension of the pseudopodia in regular intervals biased by the gradient of the chemoattractant. In (b), the **polarisation** is displayed, as the cell would turn towards the direction of the gradient. Finally, the **directional sensing** in the cell is indicated by the accumulation of proteins on the membrane towards the side of the cell experiencing a higher intensity of the gradient. For example, in neutrophils, a type of white blood cell, migration has been proven to be driven by sensing external chemical gradients [17].

The ranges of shapes achieved by cells are most often the cytoskeletal structures, such as the microtubules [18]. Furthermore, there is an emerging paradigm that cells' shapes underlie their functions, and that the sensing of the shape is based on processes

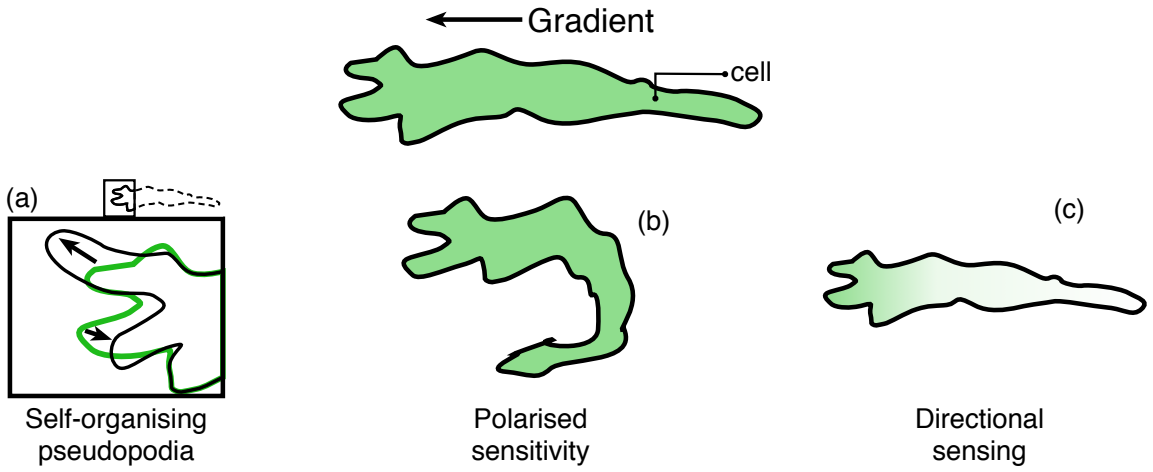


Figure 1.5 Diagram of the three interrelated phenomena in chemotaxis, as depicted by Van Haastert and Devreotes [16]. In chemotaxis, three phenomena can be observed. (a) **Self-organising pseudopodia**, which involves the movement and extension of the pseudopodia in regular intervals biased by the gradient of the chemoattractant. In (b), the **polarisation** is displayed, as the cell would turn towards the direction of the gradient. Finally, the **directional sensing** in the cell is indicated by the accumulation of proteins on the membrane towards the side of the cell experiencing a higher intensity of the gradient.

of auto-organisation [19]. Taking the work by Van Haastert and Devreotes [16] as a point of reference, the changes in morphology are directly linked to the chemoattractant increments, triggering biphasic responses. In them, the shapes of cells vary from a *resting* state, to an *adaptation* state. Figure 1.5 shows a diagram of chemotaxis as described in [16], which can be explained as the first step described in Figure 1.4(a).

Immune cells and cell migration in biological processes

In this section, cell migration in the context of biological processes is presented, especially with regards to immune system cells like macrophages. The migration of cells is of great importance in many biological processes, such as embryogenesis, wound healing and within the immune system [2, 3, 6]. In particular, the chemoattraction of immune cells, like macrophages and neutrophils, within tissues is an essential step to the host organism's response to infection [9]. The cells that form the immune system are varied, and its classification complex. The main type of cells to be referred to in this work are macrophages.

Macrophages These are cells of the immune system that filter foreign particles when settled in lymphoid tissues and the liver [2]. In homeostasis, the tendency to an

equilibrium of physiological processes, the role of macrophages range from tissue repair through to immune responses to pathogens [20]. Furthermore, clinical applications that target specific signalling in inflammatory responses are being developed [21, 22].

1.7 Mathematical modelling of cell motility

In spite of being a challenging task, mathematical modelling of cells' structures and cell motility is a large field of study [23]. This aims to produce reliable and robust models which describe the steps of the cellular movement process described before; as put by Flaherty et al. [23]. Such methods allow to explore various permutations of the same event, skipping the inherent difficulty and time associated with *in vitro* investigations. In this section, a brief overview of mathematical models of cell motility and some of its findings are presented. Such models spawn from observation and from the essential understanding of cellular movement; its objective is similar to the objective of this work. As stated by Flaherty et al. [23] "The stimuli involved and the observed response are of great interest as it would highlight reaction of multicellular entities to different environmental factors."

The review by Flaherty et al. [23] presents a classification of techniques from the biological point of view, which involve the specific parts of the cell being modelled. In the short review by Vermolen [24], a classification of such techniques in terms of the mathematical methods of modelling is presented. Two relevant approaches are discussed. First, models in which only the cell membrane is modelled as a surface in which each point in it is connected to its neighbours. The second type of model involves fully continuous models, corresponding to most of the techniques discussed in this section. In such models, the workflow usually involves a certain form of equilibrium, expressed as equations that describe physical properties of a medium, like the viscosity and elasticity of a gel, or the mechanical behaviours of cell structures, which usually exploit Hooke's laws of springs. Such equations determine its parameters, which normally are derived from the assumptions made, such as the medium's characteristics. Then, once parameters are chosen and the initial equations are simplified, a numerical technique is then used to solve or approximate the model.

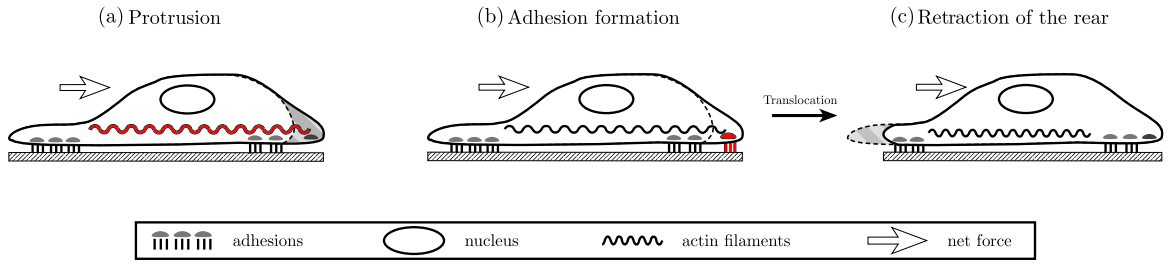


Figure 1.6 Simplified diagram of cell migration. Recalling from Figure 1.4, the relevant parts to the mathematical modelling of cells are shown: Protrusion, adhesion and retraction of the rear. The cell parts highlighted in red represent a key component in the process in the context of the mathematical modelling. (a) Represents protrusion, which is formed by actin polymerisation, and then anchoring to the substrate in (b).

To understand the key features and the reasoning behind the mathematical modelling, it is important to review the movement procedure from Section 1.6, which is depicted in Figure 1.4, which was simplified in Figure 1.6. Three major steps can be overviewed from the diagram: protrusion of the leading edge, adhesion to the underlying substrate and retraction of the rear. Paraphrasing the work by Munevar et al. [25], the leading edge of the cell is pushed forward by actin polymerisation, forming and anchor region. The body of the cell is then towed forward by contraction of the actin network within the lamellipodia, or stable protrusion areas. Subsequently, actin filaments contract, pulling the cell forward. At the trailing edge, substrate anchors are released.

As mentioned in [23], models can take a holistic approach — considering the cell dynamics as a whole process — such as in [26, 27]; or address specific mechanisms, in particular protrusion dynamics, as in George et al. [28]. Both approaches are presented with examples of their main findings.

Protrusion dynamics

The dynamics of protruding involve the development of lamellipodia, the *thin feet* caused by cytoskeletal actin protein filaments protruding at the leading edge of the cell. Mathematical models of such dynamics attempt to represent such a phenomenon [23]. Even within the models, different approaches and subsequent depth of analysis vary widely. In a relatively early model, Mogilner and Rubinstein [29] discovered that the number of filaments forming is relevant to the formation of the protrusions in the leading edge and thus to motion. More interestingly, the model showed how an excess in

the number of filaments the velocity of the protrusion will start decreasing. The pattern architecture of lamellipodia was explored in [30, 31] with different methodologies to approach the problem. Firstly, the work by Atilgan et al. [30] includes mechanical circumstances, like simulating growth or branching, into the modelling of growth, which is based on Stochastic processes. Finally, in the more recent work by George et al. [28], a more complex model is presented which attempts to describe cell deformation by modelling of the actin filaments. The model is more complex, by considering "both the mechanical and biochemical properties of actin filaments". It assumes that all deformations of the cell are a result of the cortical actin dynamics. The model includes the mechanics of the network as well as the biochemical dynamics, reflected in the consideration of the actin network as gel [28] with viscous, elastic, contractile characteristics. The dynamics — or translation from physical assumptions into mathematical form — are described in [32]. The model is of a bounded set in a 2D plane which deforms in time and the points within it. The movement is described as a function of time, and a set of equations including the characteristics of the medium are introduced. Two partial differential equations are derived to form the model, which describes the displacement of the cell by the actin filaments and the actin dynamics. The study conclusively produced a model able to describe events such as cell expansion, contraction, translation and protrusion.

Whole cell models

Whole cell models, as the name suggests, analyse the movement in a more holistic approach when modelling motion. Quoting Flaherty et al. [23], "The models used to define the movement of a whole cell range from simplified representations to highly complex models with a large number of constituents". In general, models involve the simulation of the cell membrane, as well as the simulation of the internal and external forces that act upon the cell.

Vermolen and Gefen [33] created a whole cell model of the interactions of the cell with its medium and other cells. The order in which the authors approached the problem involved an abstract model of a diffusion equation with a point source. Furthermore, a model of the membrane of the cell was proposed which accounted for the changes in cell boundary is interconnected by the cytoskeleton, modelled by springs. The authors describe it as a phenomenological model for the evolution of shape transition of cells. Further, the model describes both the displacement and deformation of the cells.

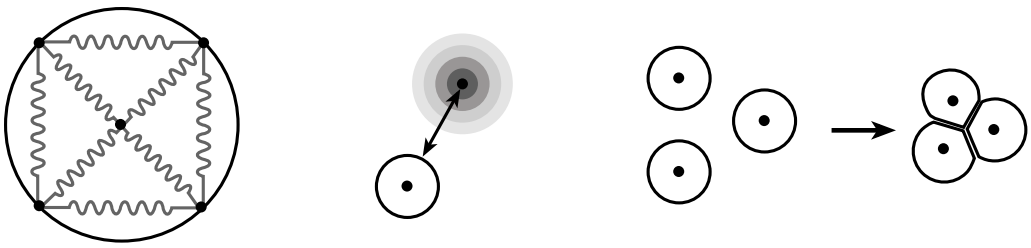
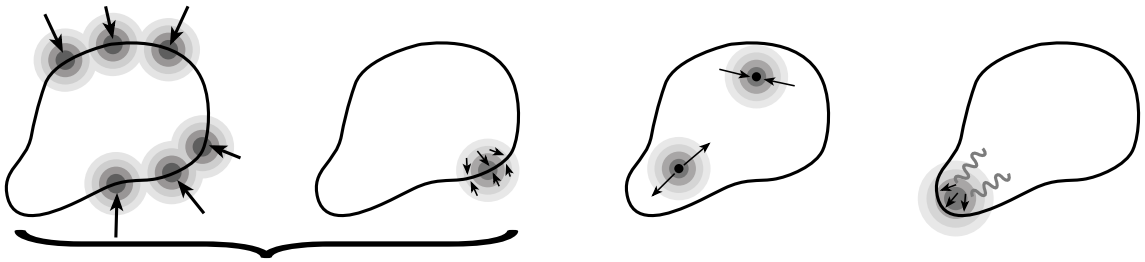


Figure 1.7 Graphical representation of a cell modelling technique based on [33]. The boundary of the cell is interconnected by springs that restore the cell to a neutral state.

Elliott et al. [26] present a computational model for cell motility, in which the cell membrane is modelled by an evolving surface, similar to [28], but in three dimensions. Three types of forces which act upon the membrane were simulated: external forces, for inhomogeneities in the medium, and pressure that constrains the closed volume of the membrane; internal forces incorporated include the reaction of the cells' surface to stiffening and bending. Finally, protrusion of the cell is modelled as a separate force through what the authors call a reaction-diffusion system. The model was solved computationally with a finite-element method. The method accounts for large deformations visible in cell migration images. The model is solved by a finite element method variation called the *moving grid finite element method*, which is thoroughly described in the work by [34], which describes in detail the numerical approach to solve the model equations.



(a) External forces (b) Internal forces (c) Protrusive forces

Figure 1.8 Explanation of a methodology of cell modelling based on [26, 34]. The diagram shows a representation of the model of a cell, modelled as a bounded region which evolves in time and is modelled as a viscoelastic medium. The different forces which act upon the cell are shown.

Another noteworthy example is the follow-up work by Croft et al. [27], where the cell motility problem is analysed in from an abstract point of view, as the core of

the research is the identification of the optimal parameters in models of cell motility, as opposed to directly modelling the behaviour. In [27], a parameter identification algorithm is presented which utilises real imaging data of migrating cells. Finally, Yang et al. [35] also presents a framework for whole cell tracking, which is relevant to this work, as it compares two different approaches: modelling of the whole cell, like seen in this section against a more traditional approach involving detection and linking the cells from frame to frame. Some of the key aspects discussed in [35] regarding the traditional tracking approach are that it does not take into consideration the morphology of the cell. It also involves added difficulties inherent to the process of detecting the cells.

Mathematical modelling discussion

As Croft et al. [27] concludes, it is typically difficult to estimate the forces associated with cell motility. Therefore, estimating parameters with such models is challenging. In [27] appears to be a point of connection between the mathematical modelling approach and the direct analysis of time-lapse videos which will be analysed in this work. The approaches compared in the work by Yang et al. [35] showcase some of the key differences of the approach taken in this work compared to mathematical modelling of the cells, whether holistic or specific. Mathematical modelling of the whole cell structure attempts to reconstruct whole cell morphologies hoping that the physics encoded in the model is reflected in the simulated data. On the other hand, some difficulties associated to mathematical models of the cells include the difficulty in choosing the parameters of the model, as discussed in [27]. Secondly, the procedures are often complicated to implement and obtain results. Finally, computational power required to run some of the models requires specialised hardware.

In this work, the difficulties of dealing with cell shapes and interactions will be addressed via a data-driven approach through geometry-based image analysis. From the images of macrophages, some empirical observations could be measured and categorised into more robust and comprehensive models. Regarding cell movement, as it will be addressed in the following sections and chapters, some interactions between macrophages are not yet fully understood and could not be immediately combined into a robust model; thus a robust method to analyse the geometry of the shapes and trajectories of the cells could provide insights into the modelling of cell movement.

1.8 Acquisition of images of migrating cells

In this work, imaging will be understood as the process of acquiring images [36], which will be analysed at a later stage. The acquired data can be single frames (photos) or time sequences (videos). Imaging allows the acquisition of high spatial resolution and high temporal resolution, allowing high detail and several frames per second to be obtained [37]. This section provides an overview of cell migration imaging as a way to visualise spatiotemporal dynamics, producing experiments that can produce data in the form of single frames, time sequences, 3D volumes or time sequences of 3D volumes. Modern immunology has been widely influenced by the visualisation of dynamic interactions between immune cells through the use of time-lapse video microscopy; although reliable quantification of said processes is necessary to draw robust conclusions [38]. Time-lapse microscopy refers to the acquisition of image sequences of microscopic objects and is reviewed in detail in Section 1.8.

Microscopy

Microscopic techniques allow for the visualisation of objects which go beyond the resolution limit of the human eye and the visible range of light wavelengths [400 - 750 nm] [1], which involve colours ranging from red to violet. The human eye requires contrast to perceive details of objects [39] and cells are generally transparent. For this purpose, several methods have been proposed: colour staining, in which a chemical is added to the sample which absorbs light at a certain wavelength, thus improving contrast; dark field, which exploits the scattering of light on small particles that differ from their environment; phase contrast, where light incident on the sample is split in two spaced beams of light; finally, fluorescence microscopy, which will be described in detail as it is relevant to this work.

The basic principle of the different techniques to acquire images from a microscope involves an energy **source** which acts upon a **sample** by scanning or transmitting through it. The majority of the images in this work, the images in this work were acquired through confocal microscopy, where the working principle involves two pinholes to restrict the passage of light at two key points.

Fluorescence microscopy

Fluorescence microscopy differs from the classic principle of microscopes presented before. The guiding principle in this case is the emission of light by the sample, when

excited with a particular wavelength. Figure 1.9 compares the visible range of sizes observable by the unaided human eye with the visible range of sizes observed by most applications of fluorescence microscopy.

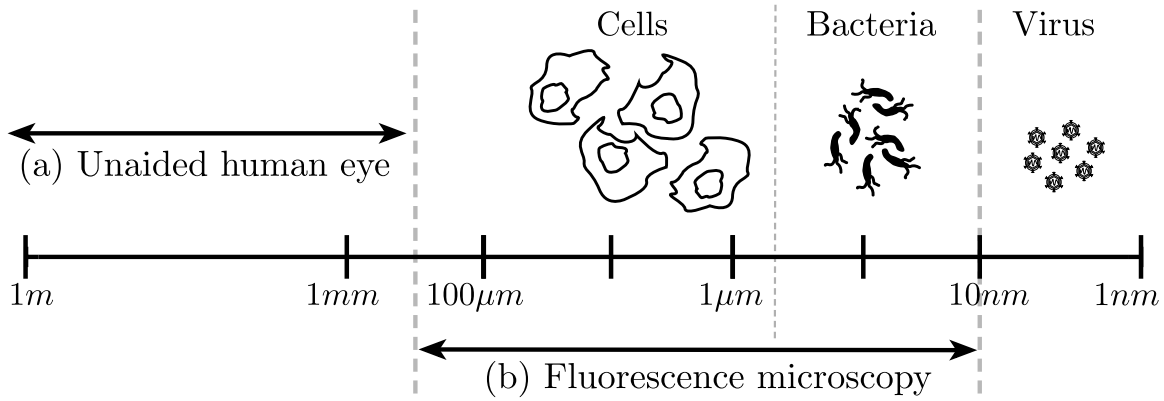


Figure 1.9 Overview of fluorescence microscopy and the size scales visualised in a logarithmic scale. The visible scale is represented, ranging from meters to nanometers. Along the scale, the visible range for the unaided human eye is marked (a), as well as the range visible thanks to fluorescence microscopy (b). At the top of the figure, examples of objects of interest within the $[1nm - 100\mu m]$ are shown for visualisation purposes. The diagram shows a modification of a schematic found in [1, Chapter 1].

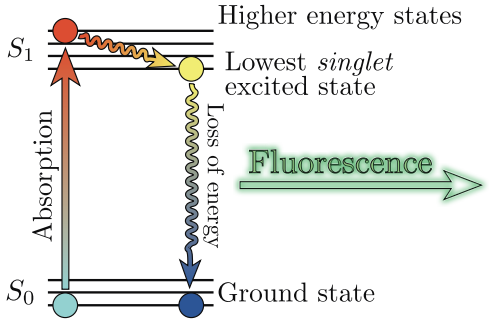
Most images studied in this work correspond to images acquired through fluorescence microscopy techniques. Fluorescence involves exploiting a characteristic in specific molecules to emit light when excited in certain conditions. *Fluorophores*, molecules able to fluoresce, can be inserted in organisms causing the effect without interrupting with the structures formed. This section explains the principle of fluorescence, and how it can be exploited to produce fluorescent samples. Then, confocal fluorescence microscopy is overviewed.

Fluorescence is the process where fluorophores, being either molecules or nanocrystals, absorb photons at a specific wavelength, and later emit them. The absorption causes electron to be excited. In order to return to ground state, there is an immediate emission of photons of a higher wavelength. In particular, fluorescence microscopy samples do not with the white spectrum, but only specific windows utilising filters. Such light filters allow the emitted photons to reach the objective.

The basic principle of fluorescence involves the change of state of the fluorescent molecule from an unstable state, caused by the illumination, to a stable one. When the change of state occurs, the excess energy is emitted as light. The atoms in a fluorescent molecule are in a stable state where all electrons are paired, known as a *singlet* state,

S_0. Upon absorption of light, change of orbitals of electrons which puts the molecule in an unbalanced excited *singlet* state, S_1, as the charge distribution of the electrons around the atomic nuclei is altered compared to the ground state. The higher state S_1 is not stable for a long time, and when the molecule returns to stability, emission of a fluorescence photon occurs. The diagram in Figure 1.10(a) illustrates the basic principle of fluorescence, while Figure 1.10(b) depicts a confocal fluorescent microscope setup.

(a) Fluorescence diagram



(b) Confocal fluorescent microscope

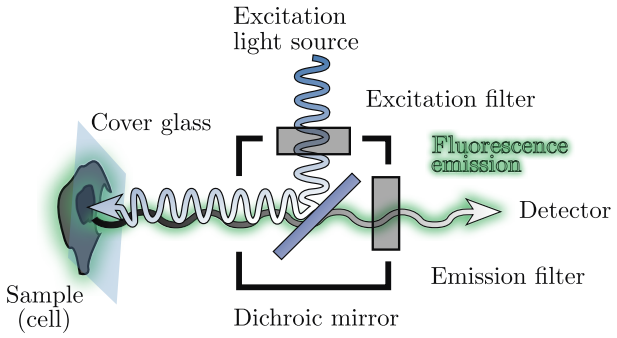


Figure 1.10 Diagram of molecular fluorescence and setup of a confocal fluorescent microscope. (a) Represents a Jablonski [39, Chapter 3] diagram in which the two states of the electron are represented: S_0 represents the ground (stable) state of the electron. When the molecule is illuminated, the particle will absorb energy until reaching a higher, less stable energy state. The electron will inevitably return to its normal state, causing a release of energy in the form of lower energy light emission. (b) shows the diagram of a confocal fluorescent microscope. The excitation light source passes through a filter and directed towards the sample, which lies beneath a glass cover, by a dichrotic mirror. Molecules within the sample fluoresce, causing an emission of light at a different wavelength. Prior to light detection, the emitted fluorescent is filtered to reduce noise.

Fluorophores are catalogued through their absorption and fluorescence coefficients [40]. These parameters determine the wavelength, or colour, at which they should be excited and the colour the emitted light would produce. The use of fluorophores requires a specialised setup of the illumination in order to produce the required fluorescence excitation [39]. Specific light sources are required as well as filters both in the illumination and collection of the emitted light. A laser beam illuminates a sample, which excites its fluorescent molecules which is then collected. A schematic of a fluorescence microscope is represented in Figure 1.10, in which the light filters setup is visible. The combination of fluorophores and confocal microscopes provide enhanced image contrast, sensitivity of detection and fluorescent labelling.

The latter point refers to the ability to produce specific compounds which target certain proteins in the object of study. The inclusion of genetically encoded labels also allow to highlight structures in the cells without interfering with their function [37]. Thus, fluorescent microscopy involves modifying the observed organisms to provoke the illumination of certain structures when illuminated with a certain light wavelengths or colours.

Naturally occurring fluorescent proteins have been widely studied, as they allow for *in vivo* studies of intracellular processes in living organisms [1, 41]. One of the most recognised and utilised markers is the Green Fluorescent Protein (GFP). The works by Zimmer [41] and Remington [42], thoroughly describe the history, structure and most common applications of the protein. In the scope of this work, Stramer et al. [4] acquired fluorescently labelled macrophages using a confocal microscope and two probes, which highlighted two cytoskeletal structures: nuclei and microtubules. The full acquisition protocol, outlined in [43], produced images like the ones shown in Figure 1.11.

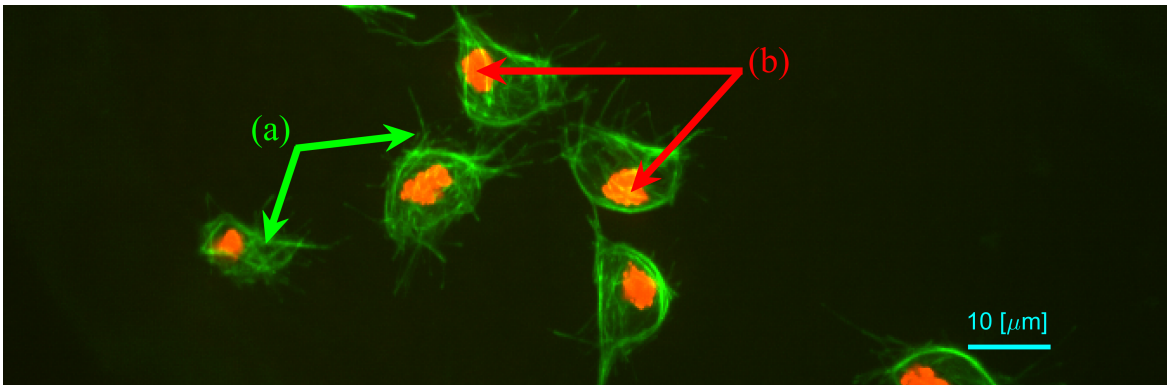


Figure 1.11 Images obtained by Stramer et al. [4] with two probes. (a) Represents the microtubule structures of the cells. Notice the green intensities are not homogeneous, thus a clear outline of the cell is not shown in the images. Furthermore, close to the edge of cells, various small structures are present in the images. (b) Represents the red probe which makes the nuclei visible. Bars: $10\mu m$.

Model Organisms

Research using animal models is a major achievement in modern medicine. Animals used in research to study a specific characteristic, that later can be linked to human processes, are referred to as model organisms. The use of animal models in biomedical research is essential to understanding the mechanisms that lead to disease in humans at a cellular level and even to providing systems for developing and testing new therapies

[44]. Different forms of cell migration can be observed both in *in vivo*, meaning in whole live organisms or cells, and *in vitro*, studies made outside the normal biological context such as biopsies, smears or smears. Many forms of cell migration are observed in developing embryos of different species [45]. In this section, an overview of different examples of organisms used for research is presented. Each subsection contains some examples of the organisms.

Rodents

Rodents, mice (*Mus musculus*), rats (*Rattus norvegicus*), hamsters (*Cricetinae*) and other rodents have been widely used as model organisms for understanding the normal functions of living organisms, as well as the functional changes associated resulting from disease or injury, according to Peters et al. [46]. These models are not only used to produce images, but to understand behaviours and test developing medicines. *In vivo* processes used in rodents involve skin wound healing, vascular sprouting and cancer invasion [45]. Each process is assessed in different ways through the model organism. A summary of the uses discussed in the work by Friedl and Gilmour [45] disclosed below.

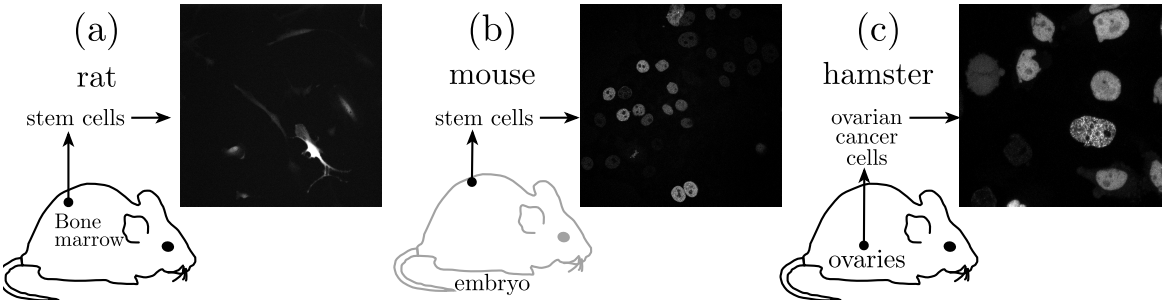


Figure 1.12 Examples of images obtained from rodents. (a) shows rat mesenchymal stem cells, usually obtained from the bone marrow, (b) shows GOWT1 mouse stem cells obtained from mice embryos and (c) shows the nuclei of Chinese hamster ovarian cancer.

Characterising the speed of wound healing was made in the work by Gerharz et al. [47], where multitissue arrays are useful tools in wound-healing studies. Multitissue arrays consist in taking a biopsy using a hollow needle removing homogeneous tissue layers. In the work by Kibbey et al. [48], an assessment of agents that stimulate angiogenesis was done by injecting subcutaneously into mice. Finally, regarding cancer invasion, the work by Alexander et al. [49] describes the routing of tumours in deep vascularised tissue.

Rodents have been also used for image analysis problems. Some examples include topological analysis of the vasculature of tumours [50], where tumours of human colorectal carcinoma xenografts were observed when growing in dorsal skin-fold window chambers in mice. Figure 1.12 displays three examples of microscopic images obtained from rodents which were analysed in this work, described in detail in Section 2.2.

Artemia Franciscana, brine shrimp

The marine crustaceans *Artemia Franciscana*, commonly known as brine shrimp, is extensively used as a model organism in studies which evaluates the toxicity in the water [51]. In common scenarios, *Artemias* is used as a test organism. In the work by Lewan et al. [52], it is also an acceptable alternative of toxicological assessments in mammalian cells. Common to most toxicological assessments, brine shrimp are submerged in controlled environments, where the water is gradually infused with a specific pollutant and the movement of the crustaceans is assessed to indicate water contamination. In [53], an experiment was recorded, and the crustaceans' movement was analysed using image analysis and tracking techniques. Figure 1.13 shows a generic overview of an experiment with *Artemia Franciscana* and an example frame.

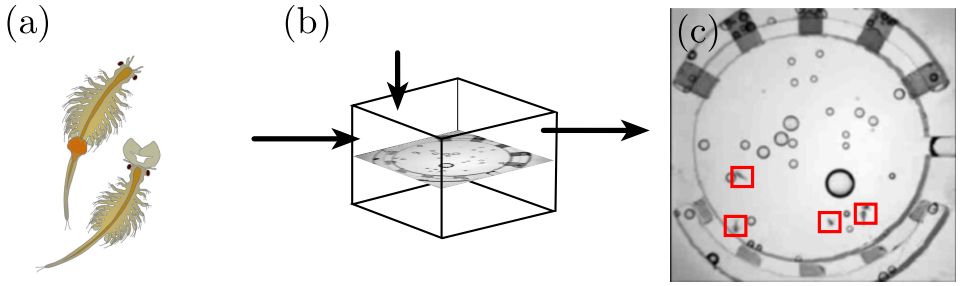


Figure 1.13 *Artemia franciscana* in traditional setup for toxicological assessment of an environment. (a) displays a figure of adult brine shrimps (reproduced from [54]); (b) is a representation of a toxicological experiment in which the *artemia* are submerged in water and the pollutant is infused gradually. (c) shows an example frame with brine shrimp swimming (red box).

Danio rerio, zebrafish

Zebrafish, *Danio rerio*, has emerged as a genetically tractable vertebrate model, which has become widely studied due to its optical clarity and embryological manipulability [44]. As concluded by Lieschke and Currie [44], the scope of zebrafish as a model

organism includes a wide range of human diseases both in early and adult life. The model organism has also particular strengths as a model for biomedical research.

For example, Zebrafish has been used as a model for the sequential process from the formation of a rudimentary organ parts or the epithelium, as discussed in the work by [55] and later in Affolter and Caussinus [56]. The regenerative characteristics of Zebrafish also allow for the study of collective cell migration. Two cases are worth mentioning, first of all the work by Aman and Piotrowski [57] which studies the molecular mechanisms of regulated collective migration, which can later be linked to understanding cancer metastasis. Secondly, the work by Haas and Gilmour [58] elaborates on the tissue cell self-organisation.

Finally, and more relevant to this work, Zebrafish allows for a detailed overview of neutrophils, a type of white blood cells. The images acquired through fluorescent labelling have enough resolution to drive image analysis research. In a case more relevant to this work, the works by Reyes-Aldasoro et al. [59] and Kadirkamanathan et al. [17] utilise zebrafish to understand neutrophil migration through image analysis. In [59], neutrophils are observed *in vivo* in zebrafish through labelling with a Green Fluorescent Probe (GFP), the images analysed provided enough information to perform shape analysis of the cells. In [17], the chemoattractant field of neutrophils is inferred through image analysis, where a transection of the zebrafish's tail would be done and the neutrophils were observed *in vivo*. Figure 1.14 represents an example of the data that can be extracted from a zebrafish embryo which has been genetically modified with a fluorescent probe to show the neutrophils.

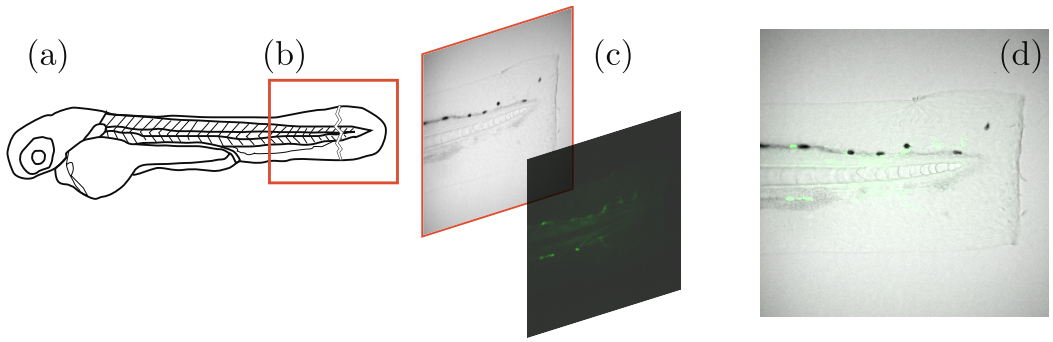


Figure 1.14 Data of neutrophils observed in zebrafish embryo. A schematic of a zebrafish's lateral view is presented in (a), with a section of the caudal fin highlighted with a red box (b) for visualising the experiment described in [17]. In (c), the phase contrast layer and the fluorescence layers of the section, and the overlapped images are shown in (d).

Drosophila Melanogaster, fruit fly

The common fruit fly, *Drosophila melanogaster*, also referred to as *D. melanogaster*, has been widely studied because on investigations of many developmental and cellular processes common to higher eukaryotes, including humans [60]. Models obtained from *Drosophila melanogaster* and zebrafish have produced most mechanistic insights as they combine genetics with *in vivo* imaging approaches, [45]. For example, *D. melanogaster* has been studied for modelling diseases [61] and the findings have been widely documented and shared [62]. Similar to the zebrafish, the fly has been used as a model organism in cases of branching morphogenesis, where the sequential process from formation of an organ anlage and invagination of the epithelium to branch initiation and outgrowth was observed in various model organisms [55, 56].

Wood and Martin [3] showed how *D. melanogaster* are offering complementary insights into how macrophages are integrating cues into migration, whether for developmental patterning or as response to inflammation. In their work, Wood and Martin [3] discuss the general process in which macrophages can be observed on *D. melanogaster* embryos. Briefly, the creation process of macrophages and other blood cellular components, called *hematopoiesis*, occurs in two waves: during embryogenesis and during pupal development. In both instances, the process involves blood cells migrating from one end of the organism, thus populating it. Figure 1.15 shows a simplified diagram of both processes of *hematopoiesis* modified from the diagram found in [3].

On the other hand, Montell et al. [63] and later Pocha and Montell [5] have shown insights into the mechanisms for single and collective cell migration, where the link between defective migratory responses and human diseases, autoimmune and cancer, has been studied.

Examples of macrophages acquisition in two settings are shown in Figure 1.16. First, Figure 1.16(c) shows a developing *drosophila melanogaster* embryo, obtained in the context of the Cell Tracking Challenge (CT-Challenge) of the IEEE Symposium in Biomedical Imaging (ISBI) [64]. Finally, being central to the developments in this work, Stramer et al. [4] showed empirically how the interactions amongst the cells' structures, appear to "*regulate and anticipate migration patterns*", Figure 1.16(d). The imaging technique described would allow for the fluorescent labelling of both the nuclei, through a GFP-Moesin marker, as well as the microtubules, obtained through a green microtubule probe (Clip-GFP). The video sequences obtained allow for the clear visualisation of the contact between cells' structures. In some cases, a distinct change

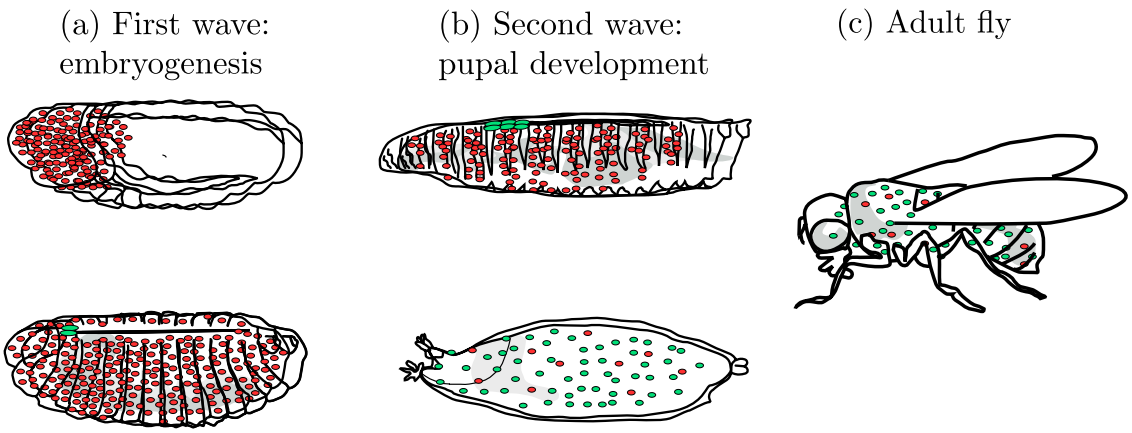


Figure 1.15 *Hematopoiesis* in *Drosophila melanogaster*. The diagram shows the two waves of cellular blood components creation in macrophages. The first one (a), occurs when the embryo of the organism is formed, in which the macrophages generated correspond to the tissue macrophages found in humans. The second one (b), occurs during pupal development, and corresponds to the fly equivalent to the bone-marrow derived macrophages in humans. The diagram is an adaptation of a diagram found in [3].

of direction in two cells which come in contact can be observed. The datasets obtained from these studies correspond to the principal data analysed in the development of this project. Section 2.2 provides a thorough overview of the datasets and the characteristics relevant to perform computational analysis.

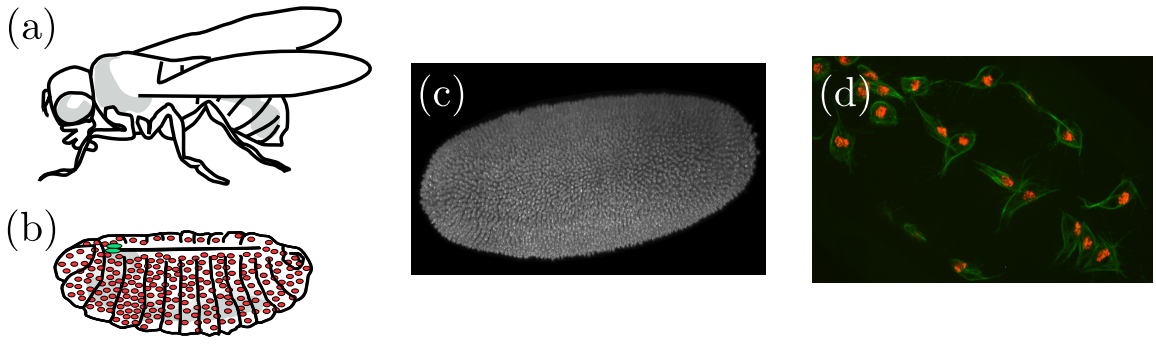


Figure 1.16 Examples of images obtained from *D. melanogaster*. (a) Represents a diagram of the adult fly. In (b) refers to *hematopoiesis* during embryogenesis as seen in Figure 1.15. Finally, (c,d) show two examples of macrophages acquired in different settings. (c) shows an image described in [64] and (d) shows an image shown obtained by Stramer et al. [4].

1.9 Summary

The general understanding is made of three main parts: cell migration in general and in the scope of immune response, understanding the acquisition of images, and an overview of some attempts made to simplify the complex biochemical and mechanical interactions through mathematical modelling.

The cell migration process is an intricate one, which encloses several sciences. Cell migration in general involves a process which in reality involves biological understanding of structures, chemical reactions happening within the cell and outside of it, as well as the physics involved in the motion. Then, observation of cells span from the technicalities of observing phenomena in much smaller spaces, to the intricacies associated to using surrogate models and linking them to benefit the understanding of humans' systems and pathologies. These should be included as not all techniques show every component in the cell, but each microscopy technique has limitations in the representation cellular structures.

The objective of such compilation was to understand cell migration as a process, as a series of steps which involve some cues and triggers and reactions. However, regardless of the vast understanding of the motion itself, this chapter aimed to acknowledge the lack of understanding the integration of such cues and, in simple terms the *intention* of the movement. Especially in the context of immunological response, understanding cell processes requires analysis in normal and perturbed conditions; ideally correlating movement and shape changes with with the dynamics of cellular signalling, both internal and external [37]. The mathematical models have been helpful in understanding and producing repeatable experiments.

In a more particular scope, the model organism *Drosophila melanogaster* can offer insights into how macrophages integrate cues to migration and other tasks [3]. It has been shown that interactions amongst the cells' structures appear to anticipate the direction of migration [4]. And therefore suggest the need for a precise analysis of cell shapes as they evolve through time as well as the correct identification of interacting cells that overlap could provide information for specific cells in biological studies; where sharp corners suggest an active migrating cell and rounded corners inactivity [4].

Chapter 2

Materials: images of fluorescently labelled macrophages and other data

This chapter describes the images analysed in this work. It is a continuation of the biological context described in Chapter 1, which concluded with the acquisition of images of cells from model organisms. The acquisition of images result in a digital array of values, representing the intensities of light captured at each position.

The following main topics are presented: (i) the mathematical foundations for the representation of images; (ii) the different datasets to be analysed in this work, where A thorough description of the images acquired is provided and the challenges to analyse the images are presented; (iii) the representation of shapes as closed boundaries of objects and the synthetic data developed to model the shapes and overlapping of the cells; finally (iv) the development of the routines to select the ground truth to validate algorithms in future sections. Two types of synthetic data were developed, allowing to validate algorithms presented in future chapters.

2.1 Representation of images

Light intensity images are the most common type of images encountered in daily experience [65], such images represent the the variation of light intensity in the scene. In analysing images of cells, as mentioned by Meijering [66], "the abundance, heterogeneity, dimensionality and complexity of the data acquired by modern imaging experiments rule out manual image management". In the scope of this work the

light intensity variation is produced by the fluorescence microscopy imaging technique described by Stramer et al. in [4, 43]. The main objects of study in this work are video sequences, which correspond to sets of time-related digital images, or simply called images. The objective of this section is to provide a homogeneous representation of digital images and the operations that can be performed in them; the overview includes the notation that will be utilised throughout the thesis.

An image can be seen as a digitalisation of a continuous function in two variables [67], corresponding to the spatial coordinates in an image. Let $f : \mathcal{I}_X \times \mathcal{I}_Y \to \mathbb{R}, f(X, Y)$ represent a function in two continuous variables, (X, Y), over two intervals, represented by $\mathcal{I} \subset \mathbb{R}$, in the Cartesian plane. The intervals can represent a rectangular section of the plane. For simplicity, and without loss of generality, let $\mathcal{I}_X = [0, a]$ and $\mathcal{I}_Y = [0, b]$, where a and b are numbers such that $a, b > 0$, positioning the rectangle $\mathcal{I}_X \times \mathcal{I}_Y$ in the first quadrant of the plane. The range of f is assumed to be the range of values $[0, g]$.

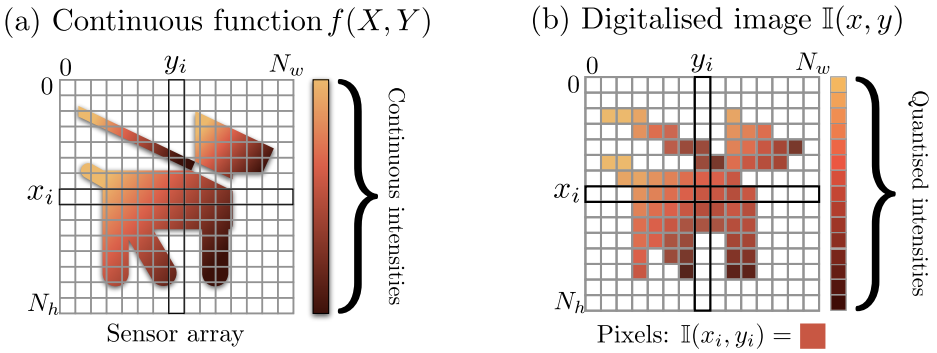


Figure 2.1 Representation of the creation of a digital image. (a) A representation of the continuous function f, the grid represents a sensor array in which the image will be projected. After digitalisation, (b) shows the function f digitalised as a matrix $\mathbb{I}$, given the sensor array. In both cases, the order of the array is presented.

The digitalisation of f consists of (i) sampling, (ii) discretisation in the (X, Y) variables; and (iii) quantisation, which is a discretisation of the range of f. Let the discrete variables (x, y) correspond to the spatial positions in $\mathcal{I}_X \times \mathcal{I}_Y$ where the samples of f are taken. For simplicity, $x = 1, 2, \cdots, N_r$, where N_r is the number of rows and $y = 1, 2, \cdots, N_c$, where N_c is the number of columns. The intensities of the image will be quantised into the finite set $I = 1, 2 \cdots, L$. For example, in 8-bit images, the quantisation results in the interval $I = 1, 2, \cdots, 255$. Throughout this work, all digitised images will be referred to with the symbol $\mathbb{I} = \mathbb{I}(x, y)$. Figure 2.1 shows a representation of the digitalisation of an image. In the figure, the arrangement of the digital coordinates (x, y) would represent the image as a matrix $\mathbb{I}$ of size $N_r \times N_c$.

Digital images are generated when an array of light sensitive sensors capture the light entering them. The number of sensors in the array corresponds to the number of positions in the resulting image. Each position in the image is called a pixel, and it is represented by the symbol $p_x = \mathbb{I}(x, y)$. The number of pixels in an image depends on the resolution of the sensor array.

Colour images

The main work carried out in this work involves the manipulation of images of colour. In broad terms, following Gonzalez and Woods [67, Chapter 6], colour is perceived by humans by the white sun light which is reflected from the objects it hits. The spectrum of colours forms the different colours which blend smoothly into each other, as the frequency of the light wave increases; however, seven colours can be broadly separated in the visible spectrum: red, orange, yellow, green, blue and violet. Figure 2.2 contains the representation of the RGB colour space, how each layer contributes to the coloured image and an example frame of the `MACROS2` dataset.

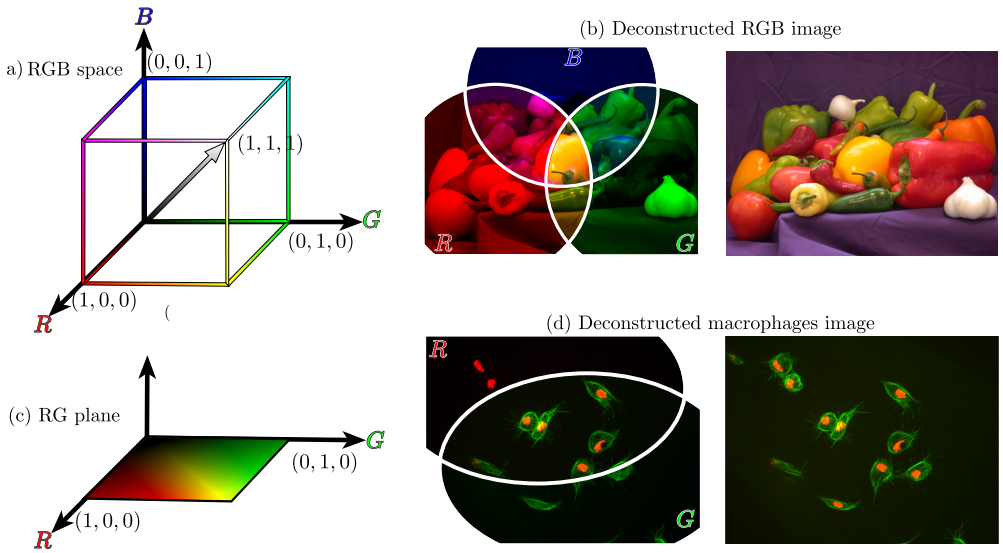


Figure 2.2 Representation of RGB colour space. (a) In the RGB colour space, each colour is represented by three coordinates ranging from 0 to 1. The greyscale is represented in the vector $\lambda(1, 1, 1)^T, \lambda \in [0, 1]$. (b) Represents the three layers in an RGB image. (c) The RGB space is reduced to the RG plane as the images in this work contain two layers: red and green. (d) Represents both layers in a macrophages frame.

For the purpose of this work it is important to understand the RGB colour space. A colour space is the specification of a coordinate system where each colour is represented

by a single point. The RGB colour space is based on a three dimensional Cartesian space in which each coordinate represents the intensity of a primary value, R for red, G for green and B for blue. In practical terms, each position (pixel) (x, y) on an image will have three values to it, each one to represent the colour. This implies that an image will be a 3D matrix $\mathbb{I}(x, y, c)$ of size $N_r \times N_c \times 3$, each of the three $N_r \times N_c$ will be referred to as layers.

Binary images

Levels of intensities in images are created in terms of a given set $I = l_1, l_2, \ldots, l_L$, in general either each $l_i = i - 1$ or they are bounded between zero and one, $l_i = (i - 1)/L$. This section presents the special case where $L = 2$, without loss of generality, $I = \{0, 1\}$. Binary images can be used to provide context to scenes in images, where regions of interest can be defined as foreground and the rest can be defined as background. In Figure 2.3, a representation of a binarised image where some some pixels are created as foreground and others as background.

The connected region in the binarised image will be referred to as detection or object, and it can be modelled as a set in a 2D plane with a size equal to the number of pixels in it. Building a binary image can be done either by manually selecting which pixels will be assigned as 1, or done automatically by a computer algorithm; which incorporates context to the image by classifying some pixels as **positive**, or "1", or **negatives**, or "0".

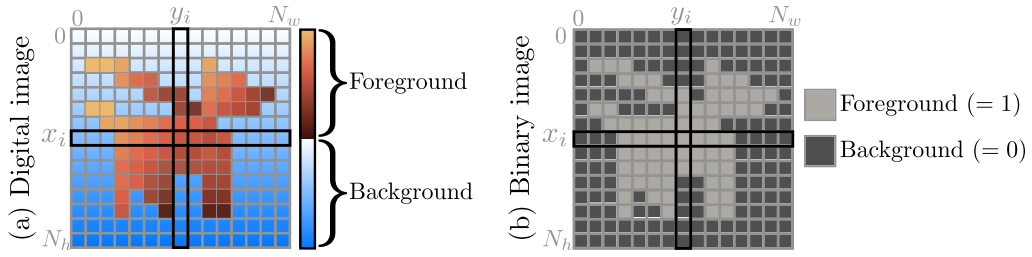


Figure 2.3 Example of a binary image defined from a digitised image. (a) Shows a digital image in which a shape can be observed and others an be appreciated as background. A binary image (b) in which the pixels corresponding to the object have been set to 1 and the background pixels have been set to 0.

2.2 Materials

This section provides the context for image analysis techniques described in the following chapters to automatically analyse movement patterns of detected objects. The following cases are presented: the ISBI Cell Tracking Challenge (CT-Challenge), the environment analysis inferred from the observation of swimming patterns in *Artemia Franciscana* and dataset of interacting macrophages. The latter presents overlapping of the cells and distinct shapes that occur during movement. To address this, synthetic data was created to aid in the testing of models and techniques developed in Chapter 4.

Datasets of the ISBI Cell Tracking Challenge

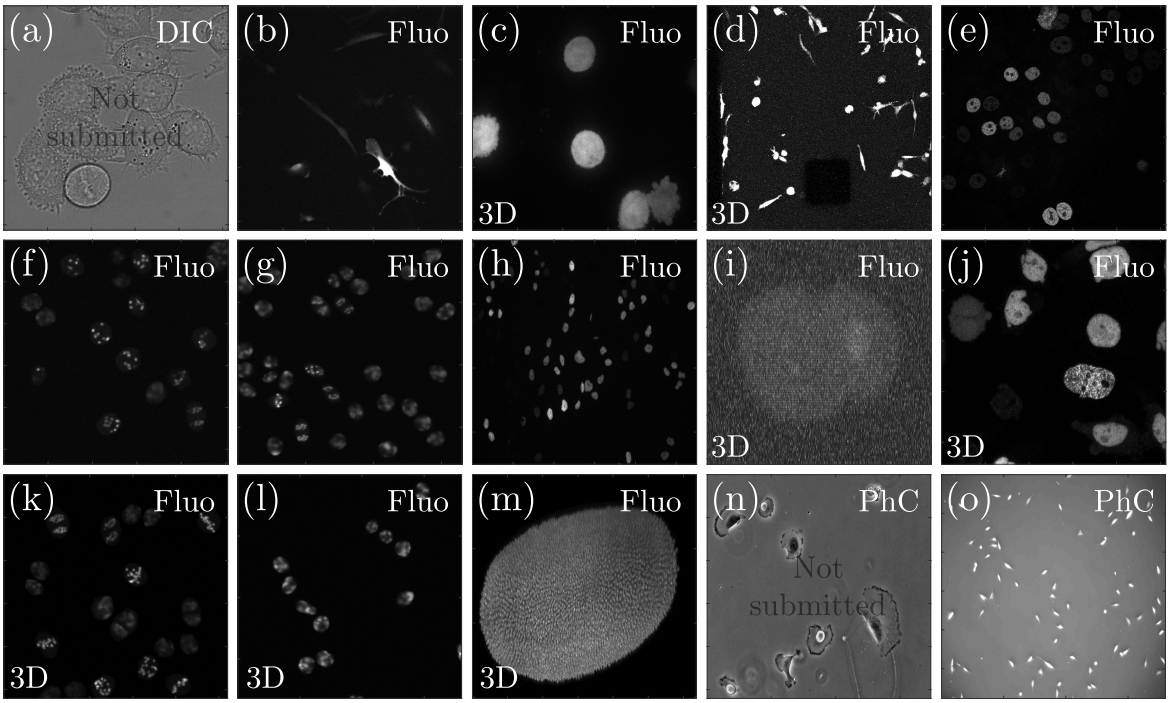


Figure 2.4 Example frames of all the ISBI datasets in the 2015 challenge. The acquisition techniques in each dataset are referred in the top right corner: DIC refers to Differential Interference Contrast, Fluo to Fluorescence microscopy and PhC refers to Phase Contrast microscopy. The top right corner shows the imaging technique. In datasets which contain 3D images display the maximum intensity projection. Dataset names are listed below, the ones submitted to the challenge appear in bold: (a) C2DH-HeLa, (b) **C2DL-MSC**, (c) **C3DH-H157**, (d) **C3DL-MDA231**, (e) **N2DH-GOWT1**, (f) **N2DH-SIM**, (g) **N2DH-SIM+**, (h) **N2DL-HeLa**, (i) **N3DH-CE**, (j) **N3DH-CHO**, (k) **N3DH-SIM**, (l) **N3DH-SIM+**, (m) **N3DL-DRO**, (n) C2DH-U373, (o) **C2DL-PSC**. Specification of the acronyms in Table 2.1.

In the context of the ISBI 2015, a combined report on the results for three editions of the CT-Challenge presented [64]. The CT-Challenge involved 21 algorithms and 13 different datasets from various microscopy modalities ranging from two-dimensional (2D) to three dimensional (3D). In the case of these datasets, the temporal context is not relevant to the work as no analysis of the tracks will be done.

Table 2.1 Description of the ISBI datasets used for this work. The table shows the basic information of each dataset. Dataset names are listed below, the ones submitted to the challenge appear in bold.

Dataset	Description
C2DH-HeLa	HeLa cells on a flat glass substrate.
C2DL-MSC	Rat mesenchymal stem cells on a flat (2D) polyacrylamide substrate.
C3DH-H157	H157 lung cancer cells.
C3DL-MDA231	MDA231 human breast carcinoma cells.
N2DH-GOWT1	GFP-GOWT1 mouse stem cells on a flat (2D) substrate.
N2DL-HeLa	HeLa cells on Histone 2B Green Fluorescent Protein (H2b-GFP) on a flat (2D) substrate.
N3DH-CE	Early *C. elegans* developing embryo.
N3DH-CHO	Nuclei of Chinese hamster ovarian cells
N3DL-DRO	Developing *Drosophila melanogaster* embryo.
C2DH-U373	Glioblastoma-astrocytoma U373 cells
C2DL-PSC	Pancreatic stem cells

The aim of the challenge, just like its predecessors such as [68], was the creation of an objective comparison between algorithms that would perform both the segmentation and tracking of the objects in each dataset. It is worth noting that none of the algorithms fully obtained solutions. Also, it is important to notice that each participating team could choose which datasets would be submitted to the challenge. For the purposes of this work, the datasets thoroughly explored included the fluorescent microscopy datasets, and one of the phase contrast.

Some of the sets have been used in previous challenges, these have been described in earlier publications [68]. Table 2.1 includes a brief description of each of the datasets that will be referred to as *ISBI datasets*, and the acquisition of the images. Some datasets are described in this section to provide context. For a thorough description, the reader is referred to [68, 64]. Examples of frames in all the datasets in the challenge

are shown in Figure 2.4. The challenge involved fifteen datasets in which different imaging techniques, differential interference microscopy (DIC), fluorescence (Fluo) and Phase Contrast microscopy (PhC) was used.

Furthermore, examples each dataset are shown in figures 2.5 to 2.9 where the difference in intensities found within the cells depicted can be appreciated. As the datasets belong to a challenge, they include an annotated training dataset consisting of the ground truth for both segmentation and tracking evaluation.

C2DL-MSC. The dataset consists of rat mesenchymal stem cells that have been transfected with Green Fluorescent Protein (GFP), the cells grow on a flat polyacrylamide substrate. Figure 2.5 shows a frame of the dataset. The dataset consists of $t = 48$ frames of size $(N_r, N_c, n_d) = (832, 992, 1)$. The dataset presents cells that in some cases appear as very thin and elongated, and in general have a low signal-to-noise ratio. Cell stretching causes some apparent discontinuities of the cells and contact between the cells. One can notice on Figure 2.5 the difference in sizes of the discernible cells. On the low centre of the image, a small bright spot can be noticed, that does not appear to be connected to other cells.

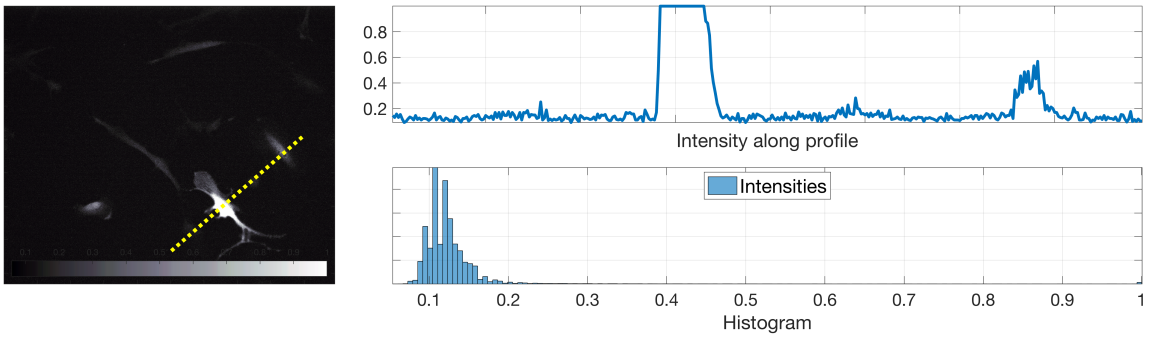


Figure 2.5 Frame taken from the C2DL-MSC dataset. The colourbar represents the varied intensities of the dataset. An intensity profile was taken and represented with a dotted, yellow line to visualise the variations in intensities in the frame. A histogram of the intensities pixel intensities is also displayed.

N2DH-GOWT1. This dataset consists of GOWT1 mouse embryonic stem cells transfected with GFP, which grow on a flat substrate. The dataset has $t = 92$ frames of size $(n_h, n_w, n_d) = (1024, 1024, 1)$. The staining of the cells is heterogeneous, which translates into a lot of variability in the intensity levels in the image, like the two cells that can barely be seen in the bottom right corner in Figure 2.6 compared to the bright

one at the top left corner. In this dataset some collisions and mitosis occur which would complicate with the tracking of the cells.

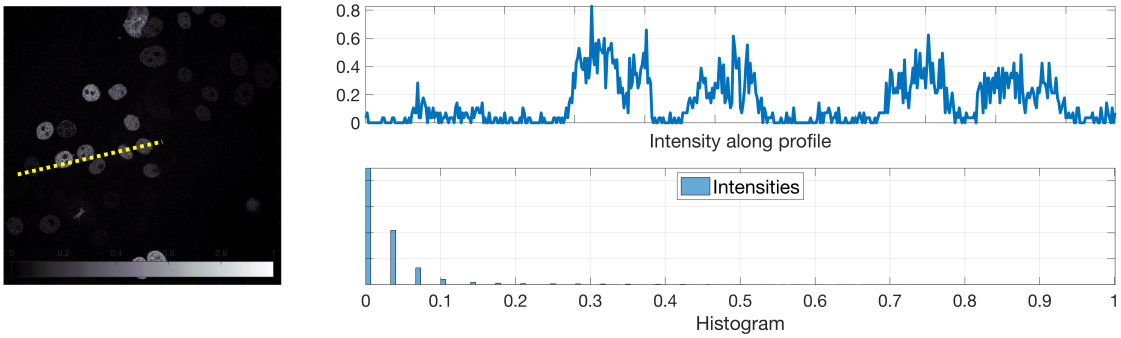


Figure 2.6 Frame taken from the N2DH-GOWT1 dataset. The colourbar represents the varied intensities of the dataset. An intensity profile was taken and represented with a dotted, yellow line to visualise the variations in intensities in the frame. A histogram of the intensities pixel intensities is also displayed.

N2DH-SIM. This is a simulated dataset of fluorescently labelled nuclei of the HL60 cell line, migrating on a flat 2D surface, as seen in figure 2.7. The HL60 cell line originated in 1977 with a patient with acute myeloid leukaemia, and have been a widely used model for studies in differentiation. For a comprehensive review of this the reader is referred to the work by Birnie GD [69]. The dataset has $t = 50$ frames of size $(n_h, N_c, n_d) = (400, 600, 1)$. This dataset includes some mitoses. The simulation tool used was based on [70].

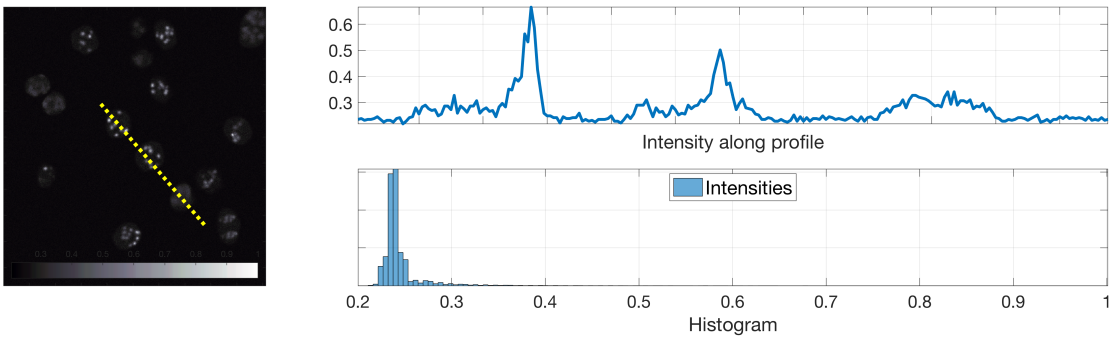


Figure 2.7 Frame taken from the N2DH-SIM dataset. The colourbar represents the varied intensities of the dataset. An intensity profile was taken and represented with a dotted, yellow line to visualise the variations in intensities in the frame. A histogram of the intensities pixel intensities is also displayed.

N2DL-HeLa. This dataset consists of Histone 2B (H2B)-GFP expressing HeLa cells on a flat substrate 2.8. The dataset has $t = 92$ frames of size $(N_r, N_c, n_d) =$

$(700, 1100, 1)$. This is a dataset with a high density and low resolution, it also displays mitoses, collisions and cells coming in and out of the field of view. The variation in intensity in the cells is noticeable as well.

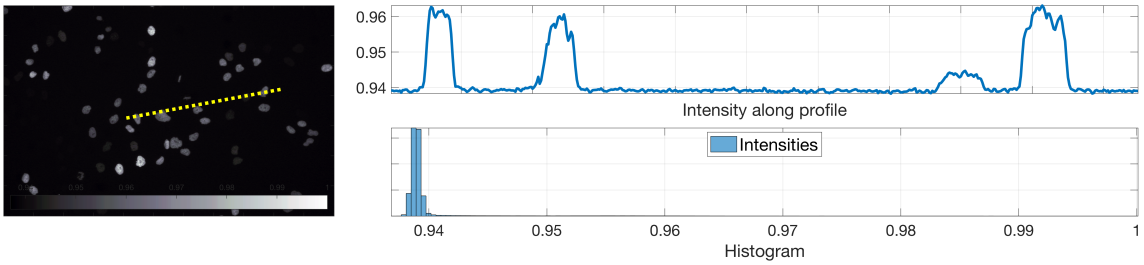


Figure 2.8 Frame taken from the N2DL-HeLa dataset. The colourbar represents the varied intensities of the dataset. An intensity profile was taken and represented with a dotted, yellow line to visualise the variations in intensities in the frame. A histogram of the intensities pixel intensities is also displayed.

C2DL-PSC. This dataset corresponds to pancreatic stem cells on a flat polystyrene substrate. It is a dataset obtained through phase contrast microscopy, which allows for a simpler microscope setup. Notice the image profile presented in Figure 2.9, in which the intensities of the background shows some shading, probably caused by the setup of the microscope.

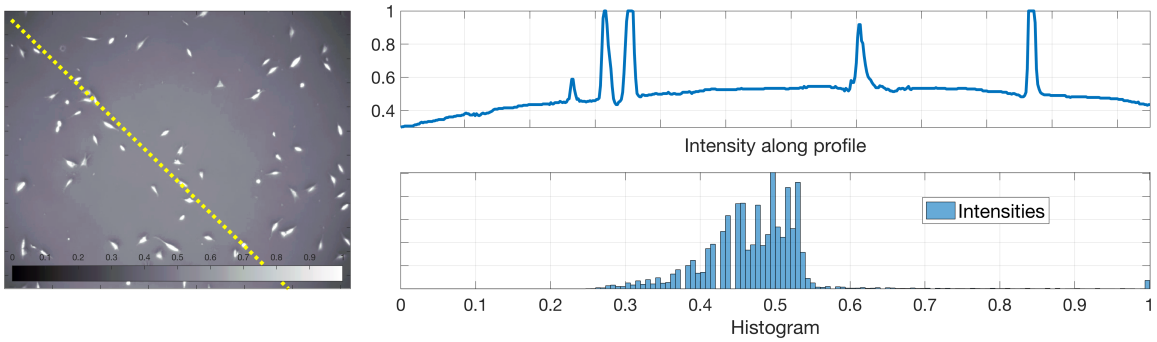


Figure 2.9 Frame taken from the C2DL-PSC dataset. The colourbar represents the varied intensities of the dataset. An intensity profile was taken and represented with a dotted, yellow line to visualise the variations in intensities in the frame. A histogram of the intensities pixel intensities is also displayed.

Artemia Franciscana for toxicological assessment

Some of the migration process of cells inside higher organisms are similar to those experienced by small organisms, in particular unicellular bacteria and also the *Artemia*

Franciscana. In [53], an analysis to of the moving patterns of the *Artemia Franciscana* was made, linking the moving patterns to the toxicity of the environment.

Artemia Franciscanas (from now referred as Artemia) are marine crustaceans that have been used as a test organism in toxicological assays. Artemia larvae are loaded into specially designed chambers where they can be observed with a USB-based digital microimaging system. The microscope is equipped with a 5.0-megapixel sensor. More specifically, the layout of the experiment can be seen on Figure 2.10, where the micro-culture chamber was optimised to hold multiple specimens of freely swimming crustacean larvae for up to 24 hours under continuous microperfusion. Microfluidic devices were perfused at a flow rate of 5mL/h with varying doses of potassium dichromate. A full description of the experiment and the materials and doses used on this experiment can be found in [53], as it is part of the work presented for the 2015 British Machine Vision Conference, in the context of the Machine Vision of Animals and their Behaviour Workshop (MVAB) 2015. The objective of the experiment was to determine, from the swimming patterns the dose-dependent immobilisation of *Artemia* larvae in response to potassium dichromate.

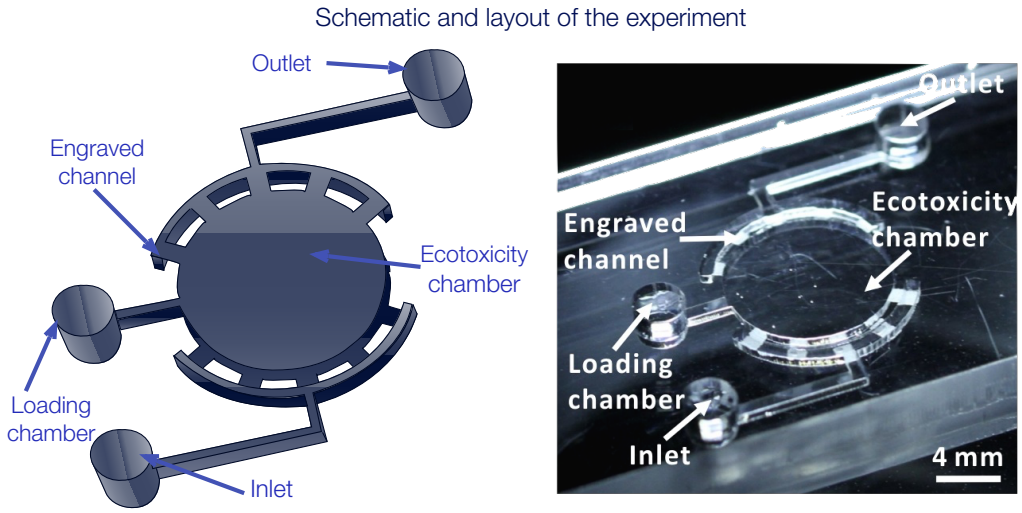
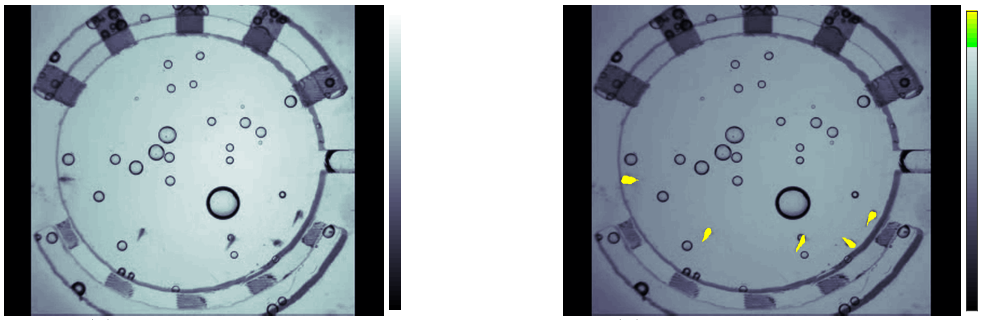


Figure 2.10 Layout of the *Artemia Franciscana* experiment. The first image (left) shows the schematic of the 3D ecotoxicity chamber of 10mm in diameter and 1.5mm in height. The second image (right) shows a photograph of the actual chamber with all its parts referenced. Image originally presented in [53].

Figure 2.11 shows an example of the dataset analysed, a $t = 1800$ time frame dataset of size $(N_r, N_c, n_d) = (288, 352, 3)$, that contains five animals and the actual objects of interest highlighted manually. Notice the presence of artefacts like bubbles

and the structure of the chamber that introduce more noise. However, if the challenges of image processing for this problems are met, the 30 frame-per-second (fps) of the imaging system allows for a very thorough analysis of the images and lets the changes of the swimming patterns of the crustaceans be better noticed and analysed.



(a) Original image (b) Ground truth highlighted

Figure 2.11 One representative image of the *Artemia* dataset. (a) Original image (b) Ground truth overlaid on the image highlighting the Artemias (yellow).

Fluorescently Labelled macrophages

Fluorescently labelled macrophages were observed in embryos of the model organism *Drosophila melanogaster*. The images show two fluorescence labels. The first layer shows the nuclei, which were labelled with CFP-Moesin, appearing red, whilst the microtubules were labelled with a green microtubule probe (Clip-GFP) [4], and appear in the second (green) channel. Three datasets were considered in this work, these will be referred to as `MACROS<ID>`, where `ID` is 1, 2 or 3. The dataset `MACROS1` was first acquired at the start of the project, and most work was developed on it, [71, 72]. The **space and temporal resolution** of all datasets consists of a pixel resolution of $0.21\mu m$/pixel with a size $(N_r, N_c, n_d) = (541, 672, 3)$; each image was acquired every ten seconds. The third layer on the RGB image corresponding to the blue channel does not contain information.

The main differences between the datasets involve the distributions of the light intensities and the number of time frames in each. Figure 2.12 show cases where cells overlap, and where the boundary of the cells would not be correctly determined. In this work, the overlapped cells are of great importance. These clusters of cells that overlap will be referred to as clumps. Figure 2.17 shows a simple diagram explaining the clumps, while examples of clumps can be found in 2.14.

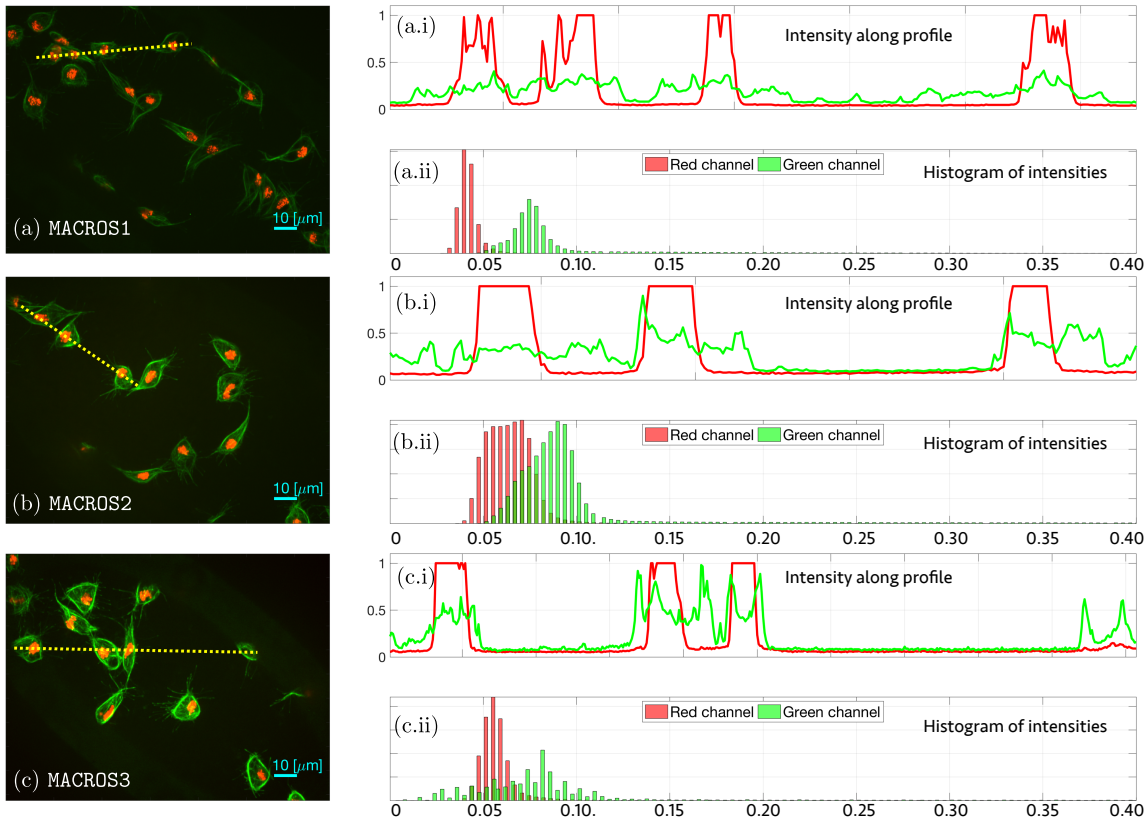


Figure 2.12 Representative time frame of the `MACROS` datasets. All datasets are represented, (a) `MACROS1`, (b) `MACROS2` and (c) `MACROS3`, where the two layers of fluorescence are visible, with red representing the nuclei and the microtubules appearing in green. Per representative frame, an intensity profile was taken in (i) a dotted, yellow line to visualise the variations in intensities in the green and red channels, and a histogram of the intensities in the frame are shown in (ii). Notice that the green channel does not fully outline the cells' boundaries, but rather presents inhomogeneous distribution of intensities. Bars: $10\mu m$.

Challenges for automated analysis of macrophages

Figure 2.12 exemplifies the distribution of the intensities per channel and dataset. Cell-cell contact, some due to possible interactions between the cells, cause overlapping in the green channels. See figure 2.14 for an example of types of overlapping between the cells. Furthermore, single cells show a range of changing shapes, examples are shown in Figure 2.15. Such shapes could be indicative of the state of the cell, as cells approaching each other appear to elongate towards ech other, as observed in Figure 2.16. The previous characteristics of the data would complicate an automated analysis of the cells through a computer algorithm. Thus, in the following section, the modelling approach for cells' shapes will be described.

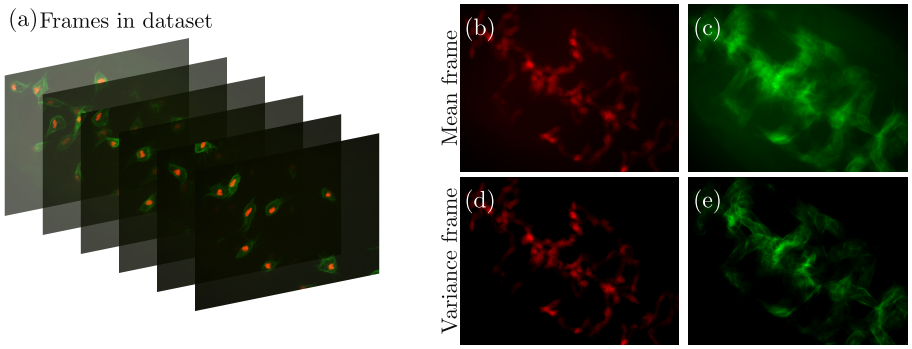


Figure 2.13 Mean and variance frames of the `MACROS2` dataset for the Red and Green channels. The mean frames for the red (b) and green (c) channels of data are shown. The mean channels were obtained by adding all the frames' respective channels and dividing by the number of frames. The variance frames for the red (d) and green (e) channels of the `MACROS2` show the areas where movement in the dataset was made. Notice that cells do not occupy the space in the top right and bottom left corners; when observing the data in motion, it can be seen that towards those edges the cells' intensities fade due to imaging problems.

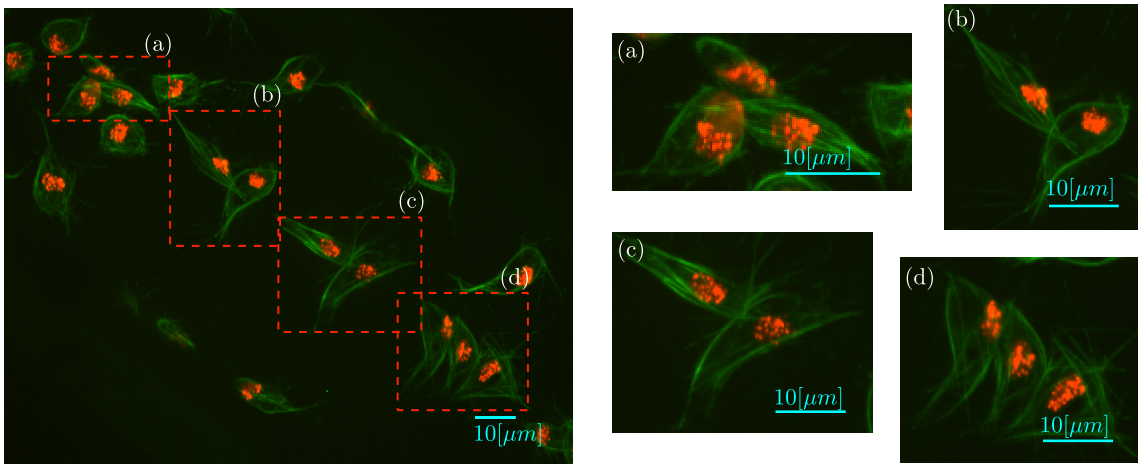


Figure 2.14 Example of overlapping cells in a single frame. On the right, representation of a full frame with (red, dotted) squares highlighting all regions of interest (ROI) with cases of overlapping in the green channel. Microtubules are clearly observed in green while nuclei of the cells is visible in red. Detail of overlapping cases can be seen in frames (a-d). Notice the different configurations of the cells, as the overlapping is caused by the interactions amongst cells. Bars: $10\mu m$.

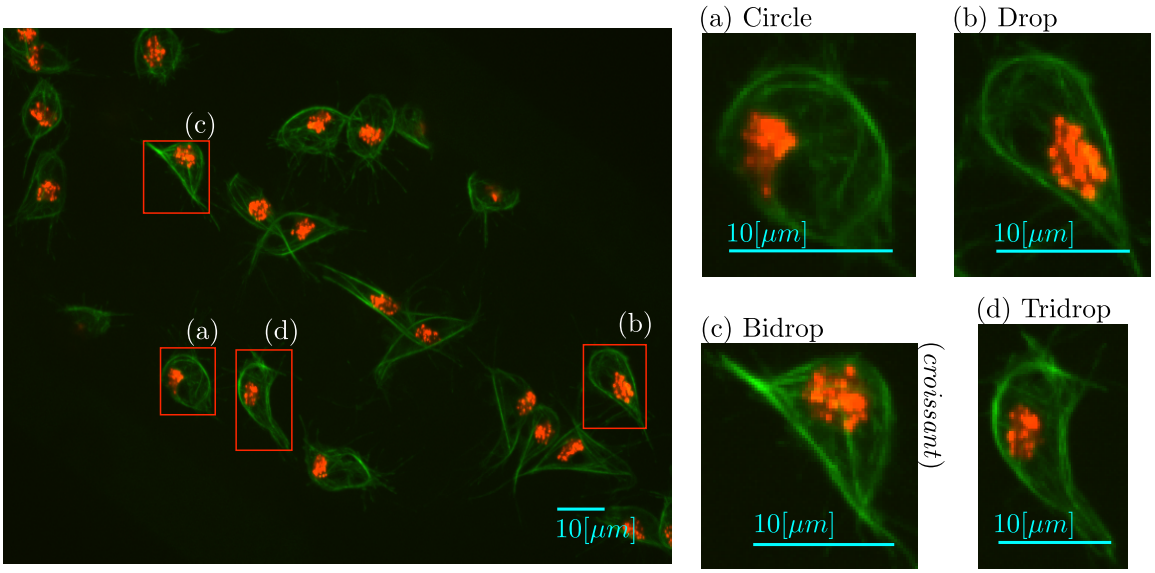


Figure 2.15 Example of cells' shapes in a single frame. On the right, a representation of a full frame with (red) squares highlighting all regions of interest (ROI) with the different cell shapes recognised in this document. (a-d) Detail of cells' shapes cases. Bars: $10\mu m$.

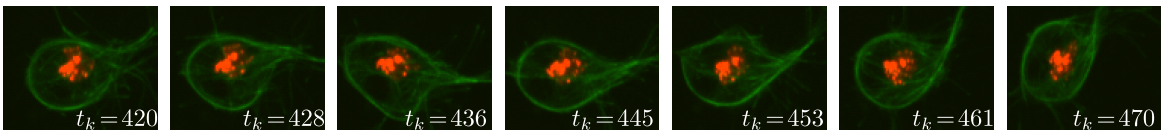


Figure 2.16 Evolution of a single cell. Seven moments selected from 50 frames. Cell has been cropped.

2.3 Object boundaries and synthetic data

Synthetic data was generated to overcome difficulties in analysing the changing shape of moving cells, as well as studying the overlapping between objects. Two separate approaches were generated to study overlapping and cell shapes respectively. This section describes the generation of synthetic data to study the two main qualities of the macrophages data: (i) overlapping cells and (ii) the basic shapes formed in the cells as it migrates.

Shapes will be modelled using boundaries. The following section defines a boundary as a closed curve, its characteristics as and discretisation are also presented. Then, a working definition of a *clump* is introduced. Finally, the synthetic data generated to model is developed. Overlapping ellipses will be helpful in studying the boundary of the *clumps* and the basic shapes will be useful in understanding the evolution of the cell shape through time, and in different scenarios, *i.e.* isolated vs. close to other cells.

Boundary of an object

Object shapes can be modelled with a parametric, closed curves, $c(s) = (x(s), y(s))$, such that $s \in [0, 1]$ and the point $c(0)$ is at the same position as the point $c(1)$. Particular to this work, object shapes are assumed to be closed and *almost* convex, *i.e.* let $A(c)$ be the area inside the parametric curve, then the curve will be *almost* convex if the area of the largest strictly convex 2D set that fits inside the curve is $\kappa A(c), \kappa \in (0, 1)$.

Discrete boundary of an object. Let $\mathcal{B} \subset \mathbb{R}^2$ define the discrete boundary an object containing N **ordered** points. Then $\mathcal{B}$ can be described as :

$$\mathcal{B} = \left\{ \mathbf{x}_i = (x(s_i), y(s_i)) \,\middle|\, s_i = \frac{i}{N}, \quad i = 1, \cdots, N \right\}, \tag{2.1}$$

where $\mathbf{x}$ corresponds to a specific vector or point with two coordinates $\mathbf{x} = (x, y)^T$. Thus, each of the points $\mathbf{x}_i \in \mathcal{B}, \quad i = 1, \cdots, N$ in the boundary are ordered. Without loss of generality, the points will be deemed to be ordered in a clockwise manner along the boundary of the object. For example, for a boundary having N points, $\mathbf{x}_{N+i} = \mathbf{x}_i$.

Definition of a *clump* The term clump, as used in this work, represents a cluster of two or more objects in a scene. A more formal definition of a clump with two objects would be defined through boundaries $\mathcal{B}_1$ and $\mathcal{B}_2$ being partially overlapped such that $\mathcal{B}_1 \cap \mathcal{B}_2 \neq \emptyset$. In an image, the image intensities would create ambiguities regarding the exact position of the boundaries of the participating objects. The ambiguity can occur in all images, whether they present various intensity levels, colour, or being a binary image. Figure 2.17 illustrates a clump as used in this work.

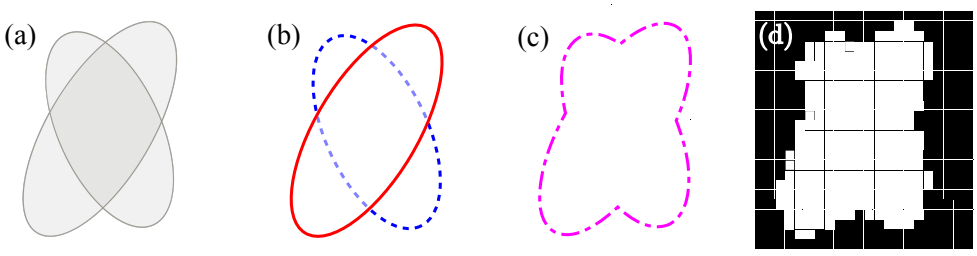


Figure 2.17 Illustration of the formation of a clump using ellipses. The diagram shows a (a) representation of two objects overlapping, then (b) presents the two different boundaries of the objects, (c) presents a clump formed by the boundary of the union of the two objects and (d) presents the binarised clump, that correspond to the mathematical union of the two binarised boundaries.

Modelling overlapping boundaries with ellipses

Let $\mathbf{x} = (x, y)^T \in \mathbb{R}$, then the parametric equation of an ellipse is given by $\mathbf{x}(s) = (x(s), y(s))^T = (a\cos s, b\sin s)^T$, where $a > b$ are the major and minor axes of the ellipse and $s \in [0, 2\pi]$. Rotating the major axis by an angle ϕ, then a rotation matrix $R(\phi)$ is defined as $R(\phi) = \begin{pmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{pmatrix}$, yielding the more general form of the ellipse $\mathbf{x}_\phi(s) = R(\phi)\mathbf{x}(s)$.

Additionally, to define an ellipse at any point on the Cartesian grid, then a translation given by the pair $\mathbf{p}_d = (x_d, y_d)^T$ would then result in the complete form of the ellipse that is considered on this work:

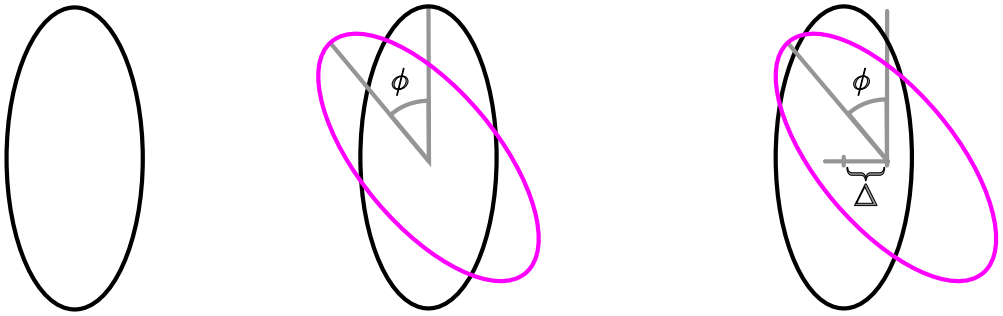
$$\mathbf{x}_{\phi,d} = \mathbf{x}_\phi(s) + \mathbf{p}_d. \tag{2.2}$$

The ellipses generated on this work follow equation (2.2). The parameters used to define the ellipses are the position $\mathbf{p}_0 = (x_0, y_0)$, the angle the ellipse is rotated ϕ, and the size of the ellipse determined by the axes (a, b). Thus, the expression of an ellipse as used in this work is

$$\mathcal{E}(\phi, d, (a, b)) = \left\{ \mathbf{x}_{\phi,d}(s) = R(\phi) \begin{pmatrix} a\cos s \\ b\cos s \end{pmatrix} + \mathbf{p}_d : s \in [0, 2\pi] \right\}. \tag{2.3}$$

The objective of the synthetic ellipses is to have a model for overlapping objects where the shapes of the objects do not vary, allowing for the study of the intersections of boundaries based on simple variables, $\mathbf{p}_d$ and ϕ.

In the objects generated using equation (2.3), three conventions were assumed in this work: The first one presents the values of the axes (a, b), meaning that $\mathcal{E}(0, \mathbf{p}_0, (a, b)) = \mathcal{E}(0, \mathbf{p}_0)$. The second convention was to have a *central* ellipse common to all pairs $\mathcal{E}_0 = \mathcal{E}(0, \mathbf{p}_0)$; this would allow the definition of the second ellipse in terms of the differences between the angles and the distance to the centre $\mathbf{p}_0$. The third convention made involves that the points taken from the centre $\mathbf{p}_0$ would only vary on the x-axis, *i.e.* if $\mathbf{p}_0 = (x_0, y_0)^T$, then for every $\mathbf{p}_d = (x, y)^T = (x_0 + \Delta x, y_0)^T$. For this work, an ellipse will be referred to in terms of the difference to the angle ϕ and difference from the centre Δx, as $\mathcal{E}_{\phi,\Delta} = \mathcal{E}\left(\phi, \mathbf{p}_0 + (\Delta x, 0)^T\right)$. Figure 2.18(a) shows the reference ellipse $\mathcal{E}_0$, while Figures 2.18(b,c) show two examples of overlapped ellipses. Once the boundaries of both objects are calculated, binary images are generated from the boundary and summed to form the test image for the proposed method. The



(a) Single ellipse (b) Overlapped ellipses (ϕ) (c) Overlapping ellipse (ϕ Δ)

Figure 2.18 Examples of ellipse boundaries generated by equation (2.2). The ellipses are generated based on these parameters: major and minor axes, an angle of rotation and translation of the x (separation). (a) Shows the single reference ellipse. (b) Shows the reference ellipse overlapped with an ellipse with a change in angle (ϕ) and in (c) with changes in angle and separation (ϕ, Δ).

binary images of size (M, N) will be represented by the variable $\mathbb{E}_{\phi,\Delta}$, this is done through the function `poly2mask` in MATLAB®, $\mathbb{E}_{\phi,\Delta} = \text{poly2mask}\left(\mathcal{E}_{\phi,\Delta}, M, N\right)$. Figure 2.19 shows one test image generated from both $\mathcal{E}_0$ and $\mathcal{E}_{50,0}$.

Generation of overlapping data Images of size $(M, N) = (256, 512)$ were generated with $\mathbf{p}_0 = (128, 128)^T$ and axes $(a, b) = (120, 53)$ that contained an overlapping of $\mathcal{E}_0$ and $\mathcal{E}_{\phi,\Delta}$. The values for the reference ellipse $\mathcal{E}_0$ are shown in table 2.2. As stated before, the images vary in terms of the separation distance Δ, and the angle with respect to the major axis ϕ. Ellipses $\mathcal{E}_{\phi,\Delta}$ were generated with values of Δ ranging from $[0, 160]$ in increments of 10 pixels, as well as angles (ϕ) ranging in $[0, 90]$ degrees in increments of 10. The images where there was no overlap present in the generated ellipses, $\mathcal{E}_0, \mathcal{E}_{\phi,\Delta}$, were discarded, producing a total of 142 images. Figure 2.20 contains a subset of the ellipses tested. Cases where there was no overlap were ignored from the analysis.

Computing ground truth of ellipse intersections In this work, the intersections of the two boundaries are recognised as key points in the boundary of a clump that would allow for its disambiguation. If the aim will be to detect relevant corners from the boundary of an object, then —at least in the context of the ellipse model— it is important to assess whether said algorithm detects the correct features.

Table 2.2 Values assigned to parameters of the ellipses to generate binary images.

Parameter	Value
Image size (M, N)	256×512
Major axis (a)	120
Minor axis (b)	53
Centre point $(\mathbf{p}_0)$	$(128, 128)^T$
Separation distances	$\Delta = 0, 10, 20, \cdots, 160$
Angles (DEG)	$\phi = 0, 10, 20, \cdots, 90$

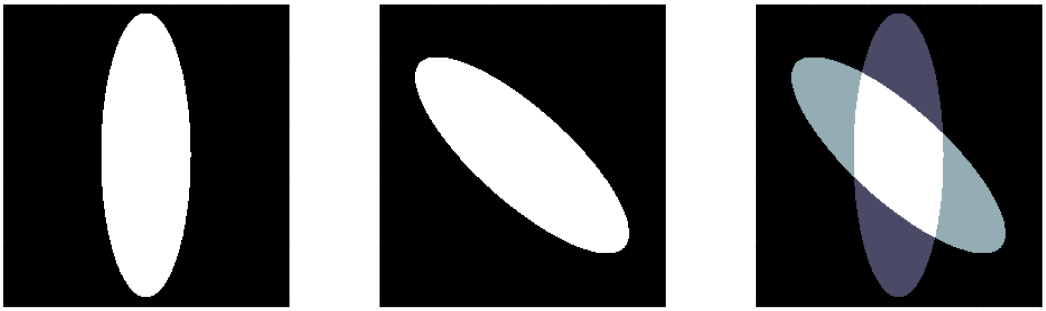


Figure 2.19 An example of the generation of overlapping binary ellipses. From the generation of various ellipses, Figure 2.18, binary images can be generated to produce a model for overlapping objects, where all their parameters are known.

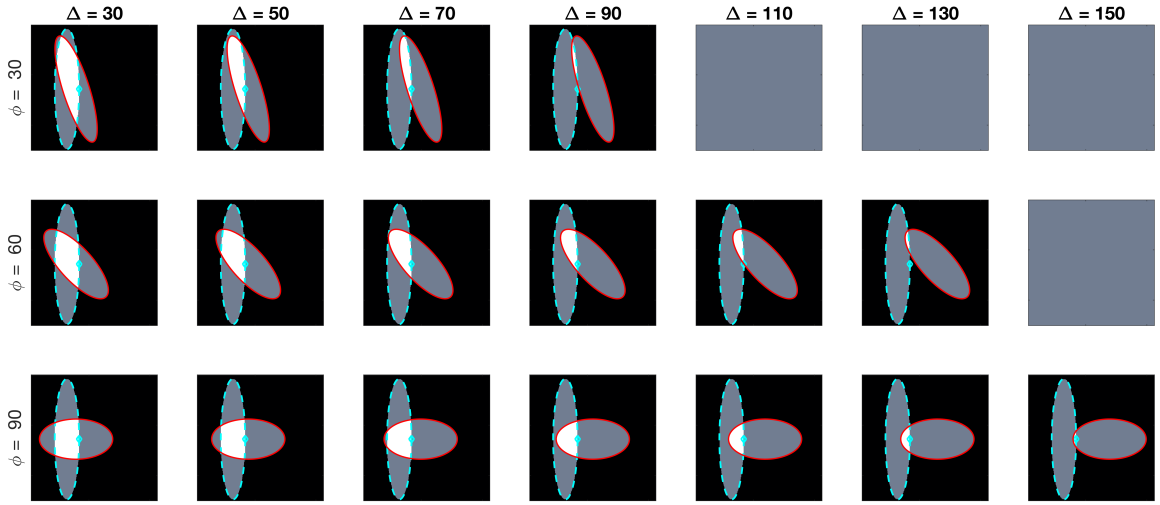


Figure 2.20 Overview of the range of paired ellipses investigated. The pairs presented on this image represent a small sample of the ellipses that were tested by the method presented. The overlapped region can be seen in white and the areas that are not overlapping are shown in grey. The boundary of the central ellipse $\mathcal{E}_0$ is highlighted in cyan (- -) while the second ellipse's boundary is presented in red.

Thus, to analyse the output of any proposed method, ground truth of the intersection points of ellipses $\mathcal{E}_0, \mathcal{E}_{\phi,\Delta}$ needs to be computed. Even though the closed solution for the equation $\mathcal{E}_0(x,y) = \mathcal{E}_{\phi,\Delta}(x,y)$ would yield the theoretical $N_{\phi,\Delta}$ intersection points:

$$X^\star = \{(x_i^\star, y_i^\star)\}_{i=1}^{N_{\phi,\Delta}}, \qquad \mathcal{E}_0(x_i^\star, y_i^\star) = \mathcal{E}_{\phi,\Delta}(x_i^\star, y_i^\star). \tag{2.4}$$

It is easy to see how the number of intersection points goes from zero to four, *i.e.* $N_{\phi,\Delta} = 0, 1, 2, 3, 4$. This approach would not be practical when more complicated shapes are needed, like the cell shapes from the ground truth of macrophages, discussed in Section 2.2. The alternative was to estimate the real intersections, which could be achieved through two methods. The first one is to analyse the boundaries of the ellipses and find the points that solve the following problem:

$$(\hat{x}, \hat{y}) = \min \|\mathcal{E}_0(x,y) - \mathcal{E}_{\phi,\Delta}(x,y)\|_2^2, \tag{2.5}$$

equation (2.5) refers to finding the set of $N_{\phi,\Delta}$ points that minimise the distance between the points in $\mathcal{E}_0(x,y)$ and the points in $\mathcal{E}_{\phi,\Delta}(x,y)$. This involves generating a matrix of distances

$$D = ((d_{i,j})) = \left\| \mathbf{p}_0^{i_0} - \mathbf{p}_{\phi,\Delta}^{j} \right\|_2,$$

where $\mathbf{p}_0^i = (x,y)_0^i$ corresponds to the *ith* point in $\mathcal{E}_0$, on the other hand $\mathbf{p}_{\phi,\Delta}^j = (x,y)_{\phi,\Delta}^j$ correspond to the *jth* point on $\mathcal{E}_{\phi,\Delta}$. In all cases, $\|\cdot\|_2$ refers to the euclidean norm.

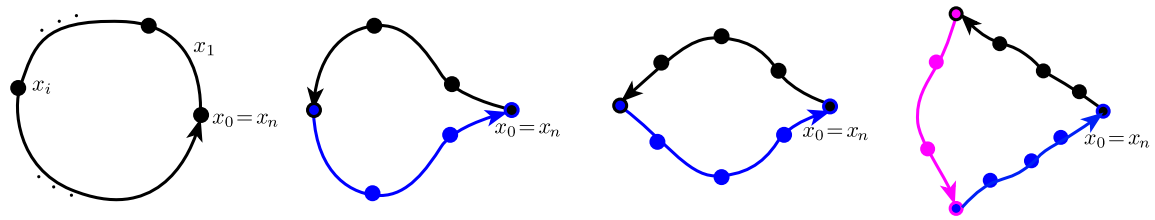
The second approach was to consider the binary images $\mathbb{E}_0$ and $\mathbb{E}_{\phi,\Delta}$, and get the binary perimeters of both, named $\partial\mathbb{E}_0$ and $\partial\mathbb{E}_{\phi,\Delta}$ respectively. They are computed through the function bwperim from MATLAB®, $\partial\mathbb{E} = \text{bwperim}(\mathbb{E})$, in this way, adding both images would cause the intersections to have a value of 2, making the finding of the intersections a trivial task. However, it is worth noting that the process is not as straightforward as the last statement, because the way the perimeter pixels are arranged in the image. The position of $\partial\mathbb{E}_0$ relative to some $\partial\mathbb{E}_{\phi,\Delta}$ could cause two potential problems: (i) not detecting all the ground truth points and (ii) having a digital segment of line intersect with another, giving more points than desired.

Modelling single cell boundaries using splines

This section describes the generation of test images that model the shape variations of the macrophages, as seen in figures 2.15 and 2.16. The shapes were generated using

control points and concatenating splines to them to achieve sharp edges with a smooth curve in between them.

Let $\{Y_i^\star\}_1^N = \{(x_{1,i}, y_{2,i})\}_1^N$, be a collection of N control points of a basic shape such that $Y_i^\star \sim \mathcal{N}(\mu_i^\star, \sigma_i^\star)$. The value of N depends on the type of basic shape: **circle** has 4, **drop** has 7, **bi-drop** has 8 and **tridrop** has 10. To model the variations in the cells' shapes, within their basic categories, the control points are distributed Normal. It is easy to see that each shape has a specific number of corners that classify them, i.e the drop has one pointy edge or corner, while bidrop and tridrop have two and three, respectively. The control points are joined with splines that then produce the **boundary** of the shape, $\mathcal{B}$, which then models that of a segmented cell (Figure 2.22). As splines are smooth curves, the pointy edges were modelled by joining different splines, depending on the shape that is being modelled. An explanatory diagram of this construction is presented in Figure 2.21.



(a) Circle: One spline

(b) Drop: Two splines

(c) Bi-drop: Two splines

(d) Tri-drop: Three splines

Figure 2.21 Synthetic generation of random basic shapes using splines. This diagram illustrates the generation of shapes using splines. The shapes that have a pointy edge would prevent a single spline to generate them, thus the concatenation of various splines was used in order to achieve the desired shapes.

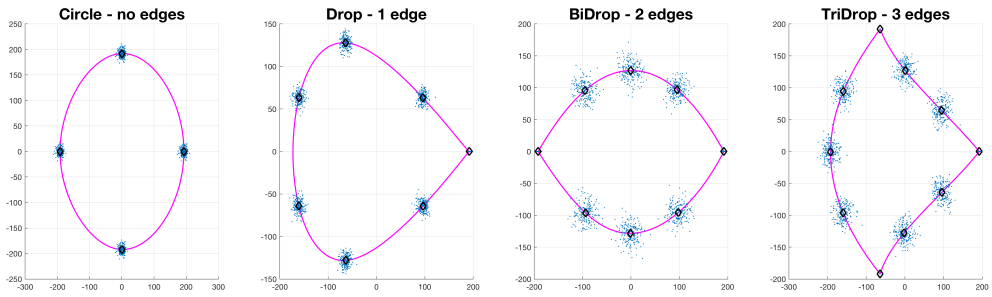


Figure 2.22 Synthetic generation of random basic shapes. Per shape, 200 cases were generated. The control points are shown in blue($\cdot$). The mean shapes are presented in magenta($-$); and the mean control points are represented in black($\diamond$).

A measure of elongation: *pointiness*. As observed in Figure 2.23 Section 2.2, the macrophages take various shapes while moving on the substrate. A key movement appears to be the elongation of the cells, especially when they are about to come in contact with other cells. Thus, the elongation is a factor of interest that would be important to include in the analysis of the movement. To model the elongation of the cell, for each of the identified shapes, a level of *pointiness* was introduced by moving the leading control points in the shape further from the centre of the figure, causing shapes where the pointy edges seem sharper.

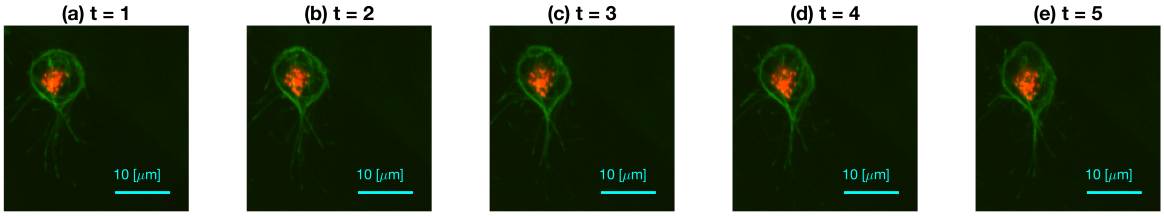


Figure 2.23 *Pointiness* changes observed in consecutive frames.

2.4 Ground truth and comparison of binary images

The ground truth (GT) is a binary image, which contains the correct labelling of objects of interest. Normally, an image is labelled manually by an expert in the field. No ground truth was provided for the data of macrophages, thus developments were made to address the selection of ground truth, which allows for overlapping cases. The GT used to accommodate this data was built through a MATLAB® software developed by the author. The GT software, which is based on MATLAB®'s `imfreehand` function, allows the user to manually label images of cells, accounting for the overlap. The user labels all cells of interest in both red and green channels of the data. Cells in the red channel do not overlap, and the ranges of shapes and sizes do not vary extensively. However, as mentioned before, the green channel presents several challenges in terms of the segmentation due to the imaging technique used to visualise the microtubules in the cells.

Generation of ground truth. Two challenges arise when developing the ground truth for the green channel: (i) varying shapes of the cells and (ii) overlapping. Because of overlapping, the software developed —`freehandSegmentation`, [72]— separates

each annotated cell onto a single layer, producing a sparse 3D matrix that contains as many layers as cells which were manually annotated. For storing purposes and handling, another representation of the overlapping data was developed which involved assigning a prime label to each annotated green channel cell and multiplying the areas of overlapping. Finally, each nuclei is given the same label as its corresponding green channel structure. The approach described ensures that each annotated data can be stored in the same space as the original RGB images. Figure 2.24 shows a diagram of the input and types of outputs available by the software.

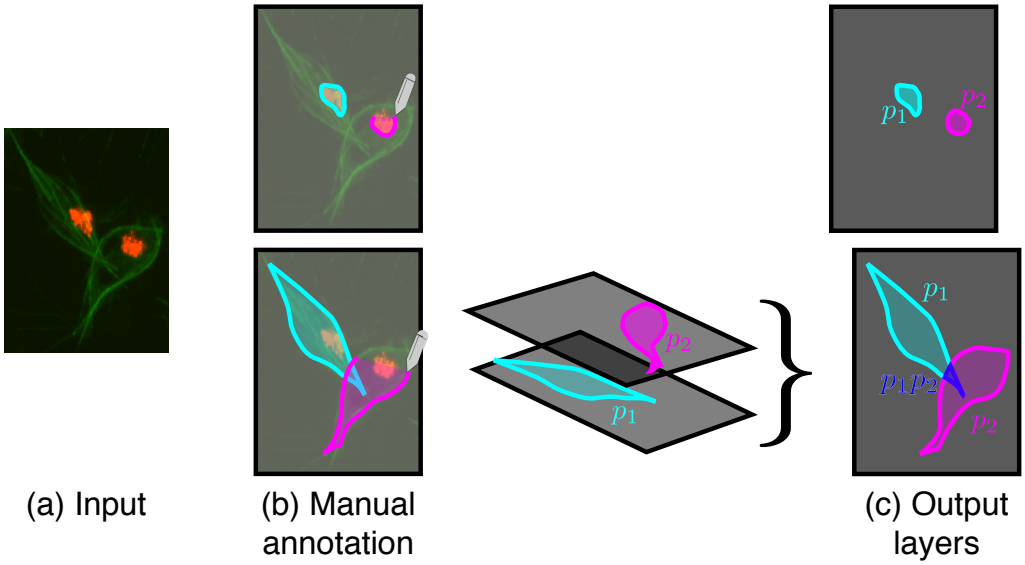


Figure 2.24 Diagram of the manual annotating software and types of annotated data. (a) The input (a) is a single RGB frame, which gets presented to the user twice for manual annotation (b). The first time, the user annotates the areas of the image corresponding to the nuclei and the second time the outlines of the green channel. Each annotation is then assigned with a prime label p_i, in this case represented in cyan and magenta. If overlapping is detected in the green channel, then each cell in the green channel is separated into a single layer while the annotation continues. Once the annotation is completed, the layers are combined into a single one by multiplying the prime labels where overlapping is present.

In this work, a **subset of ten frames** from the original 541 images were manually segmented by the author, the frames selected present examples of overlapping that can be recognised and studied, namely the four CLUMPS depicted in Figure 2.14. An example of both manually segmented channels can be seen in Figure 2.25. It is important to notice that the ground truth was generated from a non-expert's perspective. in which the shapes of the cells was easily recognisable. This part of the thesis development some of the difficulties of dealing with the dataset. As seen in the

introduction to the dataset, in Figure 2.13 the up right and low left corners are areas of low activity due to imaging problems.

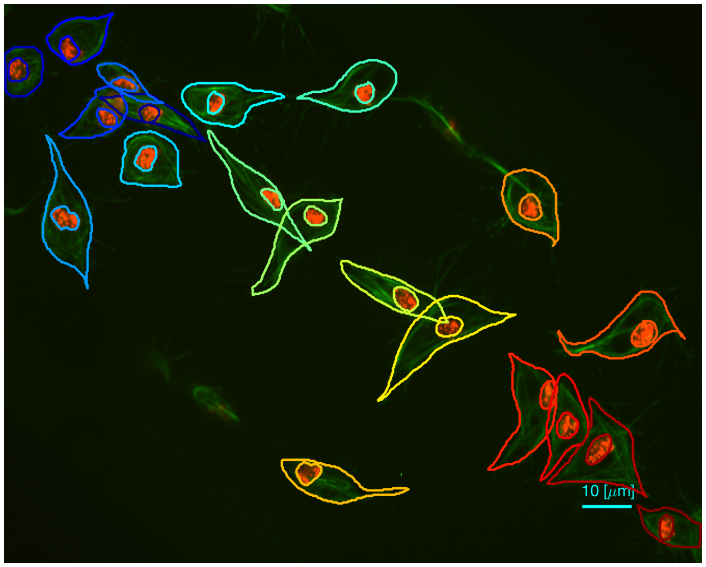


Figure 2.25 Example of the ground truth at a representative time frame. The ground truth for both red (nuclei) and green (microtubules) channels is shown in coloured lines. Both channels are visible in the image. Notice that some visible cells were not segmented, at the bottom and top. These cells were not segmented because the imaging did not provide a correct representation in the image. Notice the cell at the top, the shape of the green cell is elongated and barely visible. The case of the bottom cell, is simply because the red channel is completely lost in this case.

The ten frames selected for ground truth generation were selected for two main reasons. (i) Each frame contained four instances of clumps which could be easily identified empirically and were presented in Figure 2.14. The clumps presented 2 or 3 cells in a clump; (ii) the images come from `MACROS1` dataset, in which the contrast in the green channel is much lower thus proving to be a testing for the more challenging cases.

Comparison of binary images. In Section 2.4, the definition of ground truth (GT) was provided both in general and in the context of the data used in this work. In practical terms, given an image which contains regions to be automatically segmented, its ground truth will be a binary image of the same dimensions where all the regions were selected manually by a human, in some cases, an expert. The output of a detection software is called a segmentation, the methodologies will be explained in Section 3.1.

Manual annotations are then compared with the output of segmentation algorithms. There are various ways of comparing the annotated images with segmentation outputs,

which involve looking at images like sets. In simple terms, the analysis can be done comparing each pixel in both the ground truth and the algorithm's output image. Four possibilities, depending on the comparisons: (i) **True positive** (TP) as the pixels where both the output and ground truth show a positive or 1; (ii) **True negative**, (TN) in pixels where both images show a negative or 0; (iii) **False positive**, (FP) where the segmentation shows a 1 but the ground truth shows a 0, and (iv) **False negative** (FN) where the segmentation shows a 0, but the ground truth shows a 1.

Counting the number of pixels that correspond to each category can produce metrics for the performance of a segmentation, or how close the automatic output is from a given manual annotation, appendix B.3. In the works by Jaccard [73], Fawcett [74] and Randen and Husoy [75], such measures are provided. The Jaccard similarity index [73], or simply Jaccard index, is defined in terms of sets and their sizes, as the ratio of the intersection by the size of the union.

On the other hand, in the work by Fawcett [74], two measurements are presented: precision and recall. **Precision** measures how many detected pixels are relevant, by computing the ratio of true positives and the overall number of detected pixels, $TP + FP$. **Recall** computes the ratio of true positives with the sum of the true positives and the false negatives —or points that should have been detected, but were missed—, this measurement can be seen as a ratio of the relevant pixels that were detected. Finally, Randen and Husoy [75] present **accuracy** which compares the number of correct detections, whether positives or negatives against the overall number of pixels. All measurements can be understood from the diagram in figure 2.26, the mathematical formulation is in equation (B.26).

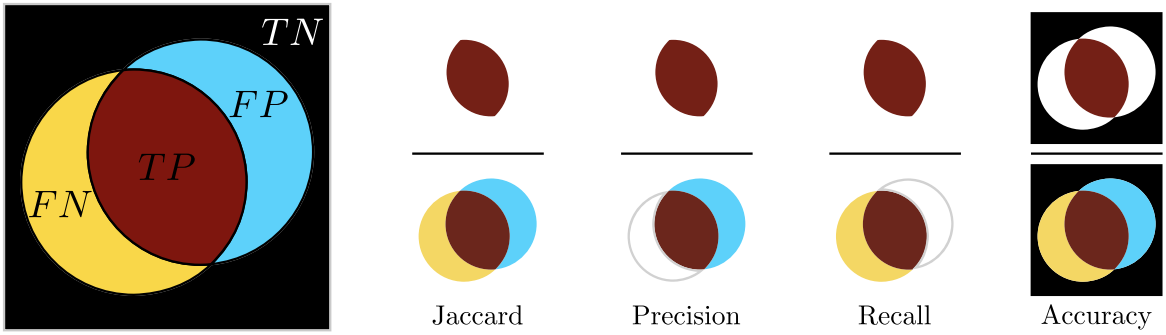


Figure 2.26 Graphical representation of the Jaccard index, Precision and Recall measurements. The diagram shows an abstract representation of the sets of pixels in an image, and how they can be classified. The ratios described in equation (B.26) is shown, based on this graphical representation.

An important part of this work involves the automatic detection of cell structures in images where the cells overlap. Therefore detections must be compared carefully to the ground truth, as the overlapping could lead to confusions. The notion of the performance measurements will be extended in the following sections to address such problem.

Comparison of ground truth in overlapping segmentations This section explains the considerations made in order to compare the ground truth manually labelled against the output of a segmentation algorithm. The software developed should consider that the overlapping nature of the green channel will likely cause that the number of cells detected by an algorithm will not match the number of cells in the ground truth. The algorithm to compare a segmentation to the ground truth is based in the Jaccard Similarity Index [73], or simply *Jaccard index*, however any statistic of performance measurement can be used in this approach. The comparison methodology involves calculating the Jaccard index of all detected cells against all cells in the output, then for each detected cell, identify which cell in the ground truth is being *targeted*.

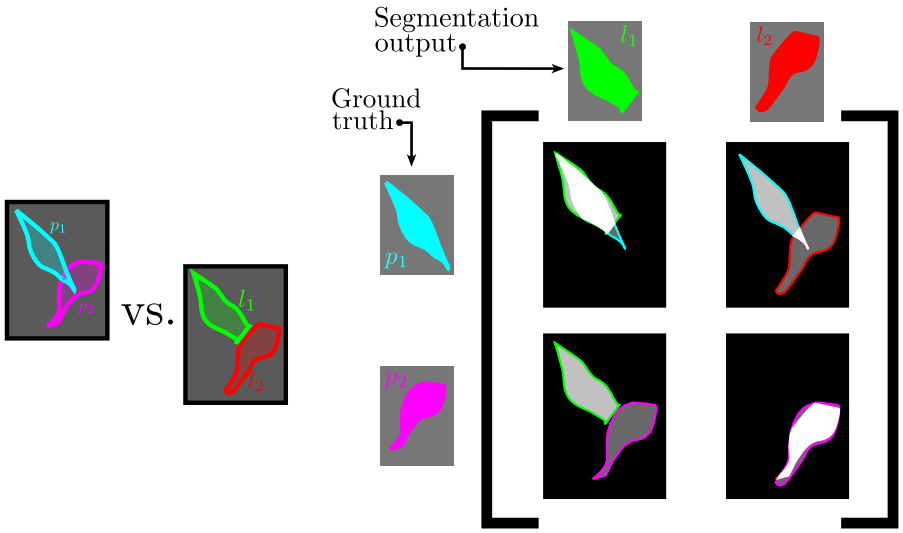


Figure 2.27 Procedure to compare ground truth (GT) of overlapping macrophages with segmentation output. Assuming that both the segmentation output and GT are organised in stacked form (Section 2.4) each cell in the segmentation output is compared to every cell in the ground truth. The diagram shows two cases of GT cells (cyan, magenta) and two corresponding cells part of a segmentation output (green, red). On the right, a representation of the Jaccard matrix $\mathbb{J}$, where the Jaccard Similarity Index [73] is performed on each of the pairs. Intersection is represented in white, which in turn involve the true positives at each case; on the other hand, different shades of grey represent false positives.

Let N_{gt} be the number of cells in the ground truth of a frame and let N_{seg} be the number of cells detected by a segmentation algorithm. The Jaccard Matrix $\mathbb{J}$ of size $N_{gt} \times N_{seg}$ contains all the comparisons from the ground truth and the detected cells. Figure 2.27 exemplifies the construction of the matrix $\mathbb{J}$. Once the matrix of comparisons $\mathbb{J}$ is built, each cell in the segmentation can be paired to the cell in the ground truth which most likely is being targeted.

2.5 Summary

This chapter presents the materials used in this project. The chapter includes of four main topics: (i) the mathematical representation of images, (ii) the main datasets of migrating macrophages analysed in this work, and the secondary datasets used; (iii) the representation of cell boundaries and (iv) the generation of ground truth.

Understanding the context of acquisition and cellular processes in the previous chapter enhanced the familiarisation of the analysis with the data. Furthermore, the presentation of the creation of digital images was of useful to understand the broad context of digitalisation and introduce some key concepts and notation, such as size of images or **colour space** and referring to channels (green or red). Segmentation, which will be addressed in the following chapter, can be defined as the creation of a **binary image** by classifying certain pixels from others.

In Section 2.2 Materials, three collections of datasets were presented, corresponding to preliminary work and main focus. Preliminary data includes the ISBI datasets and the *Artemia* datasets, while the main data consists of three datasets of macrophages with two levels of fluorescence. The ISBI datasets are an overview of the field of cell segmentation and tracking. The different datasets show a wide variety of intensities and common problems encountered in the field, such as different intensity levels, shadowing caused by the microscope, disappearing cells which could complicate the tracking and even some cellular events like mitosis. On the other hand, the Artemia dataset presents itself as an example of a project in which biological research questions can be extracted from the analysis of the movement or tracks acquired. Finally, the `MACROS` datasets were shown to give an overview of the data driving the whole project. The two levels of fluorescence appear differently from each other. If analysed separately, the two channels could provide complementary information. The nuclei (red) could be used as an overall estimation of the position of the cell, while the shape of the cells could be

investigated analysing the microtubules (green). In Chapter 3, the data is explored thoroughly.

Finally, two developments were made: modelling the most relevant features observed from cells in the `MACROS` datasets, section 2.3; developing a tool for the manual annotation of data and comparison of segmentations considering overlapping. Overlapping was modelled through ellipses, as they would allow for a controlled study of the level in which an overlapping could potentially be handled by an algorithm analysing the outline of the *clump*. On the other hand, the development of single shapes through splines allowed for a controlled elongation of shapes similar to those found in the data. Both models would allow for a controlled study of the geometry of the shapes, which will be explored in Chapter 4. Regarding the generation of ground truth, 10 frames were selected, which represented four clumps consistently moving without alteration. The number of cells in each clumps ranged from two to three and they appeared to have different levels of *difficulty*. The tool developed would allow for other applications where ground truth is not widely available.

In the following chapter, an overview of image analysis techniques applied to the data will be presented. The information empirically observed from the intensities of macrophages will be explored and quantified with the objective of finding the overall shape of the cell. Several algorithms will be tested with the datasets.

Chapter 3

Exploration of the data through image analysis and tracking

This chapter presents and exploration of the data through image analysis, showing the processes of image alteration and modification, commonly grouped as preprocessing, segmentation and postprocessing. The chapter also presents tracking as linking objects from one point in time to another, of time-sequence models.

The objective of this chapter is to provide an analysis on segmentation and tracking methods through various datasets and evaluate their performance to develop tools for an automatic analysis of data described in Chapter 2, especially macrophages in section 2.2. The exploration of the data touches on overlapping cells, in which traditional segmentation techniques produce *clumps*.

All methods presented are evaluated on the `MACROS<ID>` datasets to provide context for the reader in terms of the design choices when developing the tools to analyse the data (chapter 5); some techniques are also applied in some of the ISBI datasets, Section 2.2, and the *Artemia* datasets, 1.8.

General notation. Throughout this work, column vectors will be represented with lowercase bold letters, e.g. $\mathbf{x}, \mathbf{v}, \mathbf{p}$, and matrices will be represented either in uppercase letters, like A, B, C or in blackboard uppercase letters when an image is being referred to, for example $\mathbb{I}, \mathbb{K}$; where each element can be represented symbolically as $A = ((a_{i.j}))$, where i represents the *ith* row and j represents the *jth* column in the matrix. Values in images at a certain location, $\mathbb{I}(x, y)$, are referred to as *pixels*, and are represented by a lowercase p with a lowercase subscript, e.g p_s, p_r. The plane containing the pixels in the image will be called the *spatial domain*.

In the following Section 3.1, an overview of image processing techniques is presented, centred in particular around image segmentation. Processes performed prior and after segmentation are called pre- and post- processing techniques. The section is thus separated in all three categories.

3.1 Classical image analysis

In this work, three main stages in image processing can be recognised, which will be analysed in the following sections. The stages are preprocessing, which refer to the intensity based operations performed on an image to aid in the segmentation techniques perform better. Then, the core process of image segmentation will be discussed; such process involves the classification of each pixel into one of two categories: background and foreground. Finally, a postprocessing stage follows in which small errors in segmentation are addressed, this stage usually involves some type of binary operation. Sections 3.1, 3.1 and 3.1 show the relevant techniques in each category, applied to the data presented in sections 2.2, 1.8 and 2.2. In particular, Figure 3.1 displays the test frame which will be used to illustrate many of the techniques described herein. As the objective is comparison, a region of interest of the image will be presented, as it showcases the dataset's principal problems.

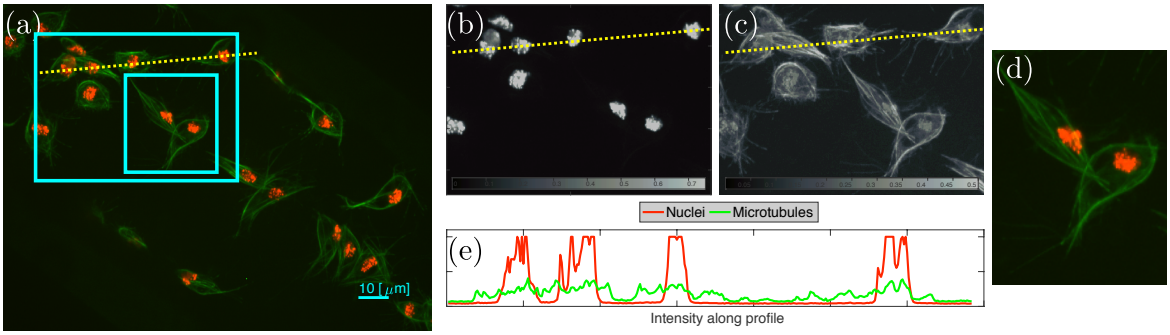


Figure 3.1 Presentation of **test frame** and the cropped images used in testing throughout this chapter. The frame will be referred to throughout this work. (a) Represents the full size frame from the `MACROS1` dataset. Cyan boxes represent the areas of detail that will be presented on the different techniques. (b,c) show the individual red and green channels. (e) Shows the colour bars: (b) $[0, 0.75]$, (c) $[0, 0.5]$. A yellow, dotted line is superimposed over some cells in (b,c) to represent the image profile presented in (e).

Preprocessing

Preprocessing of an image in this report refers to the operations performed on an image that allows for a more precise segmentation or analysis of the data. These operations can highlight some of the images' features, remove noise, modify dimensions and should improve the results of posterior operations. One classical classification of preprocessing techniques is from the work by Sonka et al. [76], which classifies such techniques into pixel brightness transformations, geometric transformations and transformations in local neighbourhoods. Pixel brightness transformations refer to modifications in the image regarding the pixels themselves, which normally address brightness or grey-scale corrections. Geometric transformations are made to address different types of distortion an image may present. Finally, local-preprocessing involves the operation in the neighbourhood of a given pixel in order to produce a new value for that location.

Furthermore, another approach in the work by Gonzalez and Woods [67] classifies the techniques into transformations of intensities and spatial filtering. Transformations of intensities, refer to transformations to the intensity levels $I = 1, 2 \cdots, L$ in each pixel, which produce modifications of the statistical distribution of the intensities. Spatial filtering refers to the local neighbourhood operations discussed in [76]. Techniques which involve the convolution of the image with a filter kernel, to produce a new image in which each pixel is the result of a local operation. In this work, the preprocessing techniques used fall into the local-preprocessing and pixel brightness categories. Figure 3.2 shows a graphical overview of the techniques analysed in this work and the classification it has in classical literature.

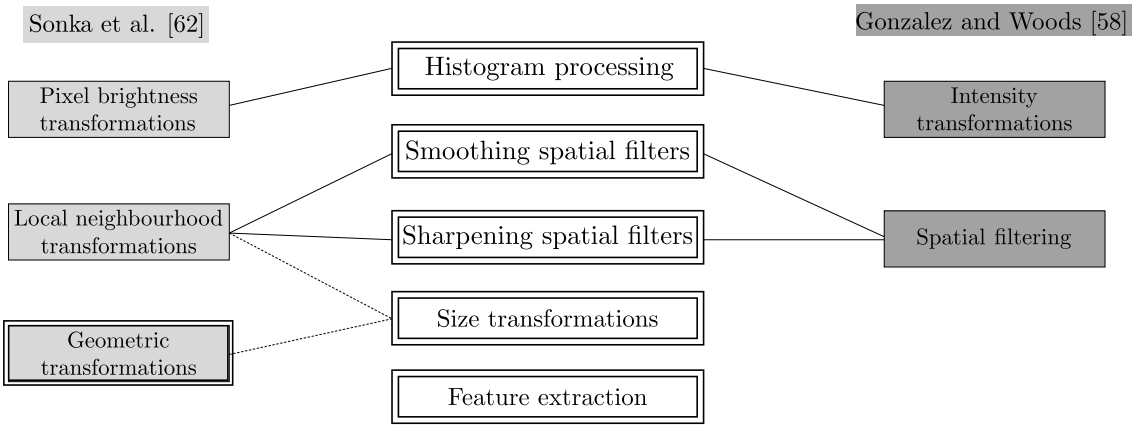


Figure 3.2 Overview of the preprocessing methodologies analysed in this work. In the centre of the diagram (double line) all the techniques as studied in the upcoming sections.

Histogram modifications

Every pixel of the image will change its own value based in a determined in a function. The transformation $T[\mathbb{I}(x, y)]$ can be rewritten as a one-dimensional function applied to each pixel $T(p)$, it is worth noting that the location is not relevant to this transformation, only the intensity level at each point. Some examples of histogram modification are described in Table B.1.

Table 3.1 Histogram modification functions. Let p_r be a pixel in an image, which can be one of the intensity levels $I = 1, 2, \ldots, L$; and let p_s be the modified value after applying the function $T(\cdot)$, $p_s = T(p_r)$.

Function on pixel	Description
$p_s = L - p_r$	Negatives, used to invert the levels
$p_s = c \log(1 + p_r)$	Logarithms
$p_s = cp_r^{\gamma}$	Power-law or Gamma corrections
Contrast stretching	Changing the range of pixel intensity values.
Histogram equalisation	Increase the global contrast of an image.

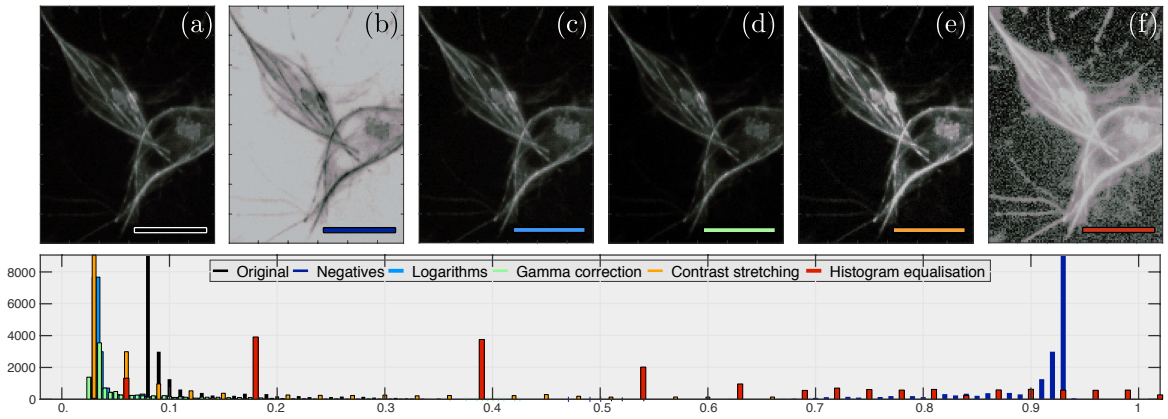


Figure 3.3 Comparison of histogram modification techniques with detail of the green channel of a macrophage image. (a) Original image. (b) Negative of the original. (c) Represents a logarithm transformation. (d) Gamma correction, (e) Contrast stretching, and (f) Histogram equalisation are more complex techniques.

Size and geometric transformations

A common transformation is to reduce the size of the image in order to increase the signal to noise ratio [77]. In their work, Burt and Adelson [77] developed an algorithm to reduce an image. Given an input $\mathbb{I}_0 \in \mathbb{R}^{m \times n}$, then the image will be reduced to an output $\mathbb{I}_1$ of size $\lceil m/2 \rceil \times \lceil n/2 \rceil$, which will be a reduced, low-pass filter version of $\mathbb{I}_0$.

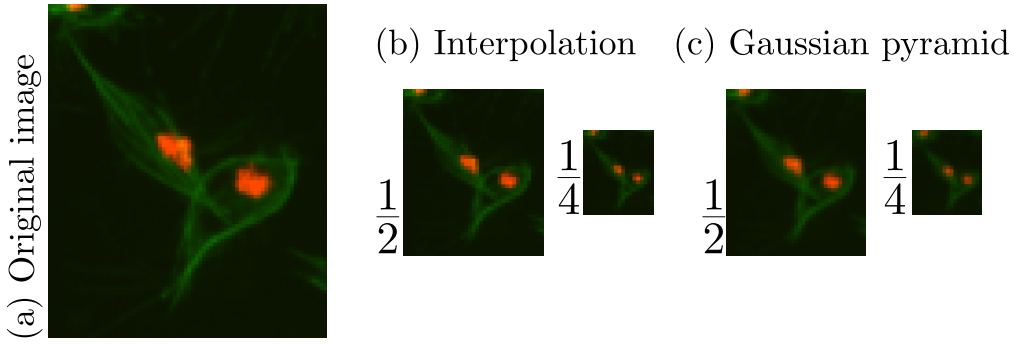


Figure 3.4 Examples of size reduction using both things proposed. From the original image (a), size reduction is performed twice, to create an image of half the size, and an image of 25% the size. (b) Shows the outputs for the image when using interpolation and (c) shows the images when using a Gaussian pyramid.

Geometric transformations are described as modifications of the spatial relationship between the pixels in an image. Linear transformations are described by a matrix $\mathbb{T} \in \mathbb{R}^{3\times3}$. Geometric transformations are shown for completeness, although they are not a part the scope of this work. Transformations can be defined via matrices which alter the positions (x, y) in an image, allowing it to change its size, the ratios of their rows and columns and translating it to a different point. The transformation matrix $\mathbb{T}$ is of size 3×3.

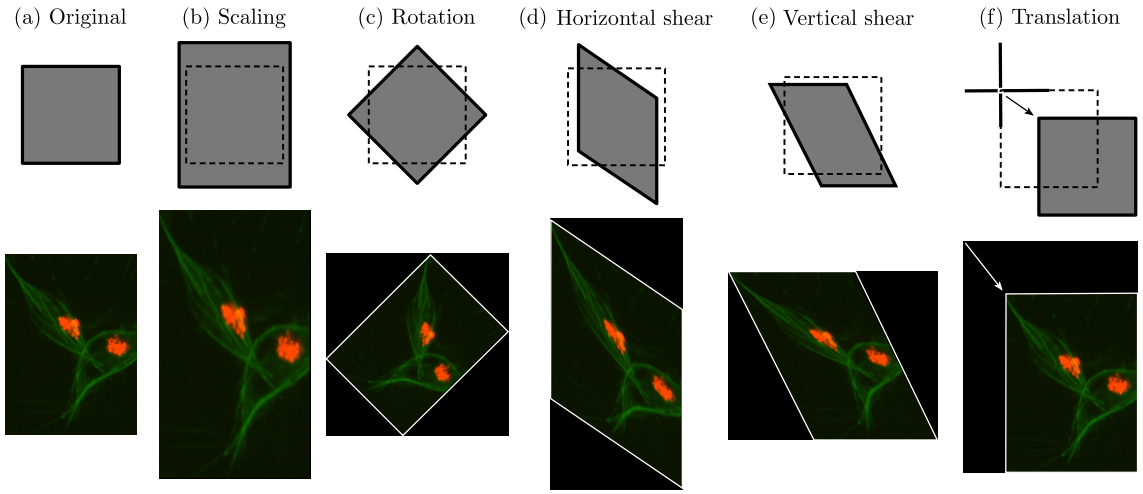


Figure 3.5 Examples of transformations on detail of a frame. The top row shows the transformations with an abstract 2D object, with the dotted lines representing the original shape. White lines have been added for visualisation purposes.

Filtering

This section provides an overview of the techniques in which each pixel in the image is assigned a specific neighbourhood in which a predefined operation is performed, [67, Chapter 3]. Filtering can be viewed differently depending on the field of study. Apart from reducing noise in the images, other filters can be used to enhance some properties or features of the image, extracting relevant information from it. The term *filtering* is taken from the field of digital signal analysis and processing in which properties of a signal are selected, enhanced or suppressed through convolution with a finite function with special characteristics. For a thorough review of convolution in 2D digital signals, the reader is referred to Section A, while for Signal Analysis and Processing, the classical works by Oppenheim et al. [78] and [67] are recommended.

Filter kernels consist of matrices commonly of small sizes that can have different parameters that determine the values within them. Throughout this work, it can be assumed that all kernels, represented with letter $\mathbb{K}$, are square matrices of size n, with $n = 2a + 1, a \in \mathbb{N}$, unless explicitly determined.

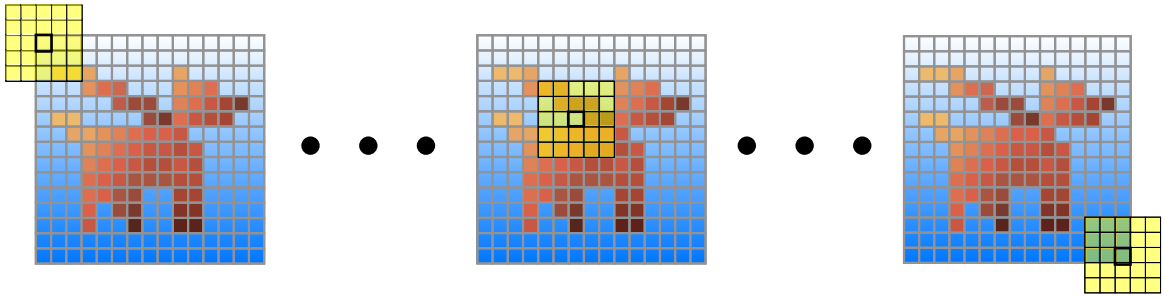


Figure 3.6 Simplified diagram of a 2D convolution of an image and a 5×5 kernel. This is a simplified version of the diagram in Figure A.1. In this case, the kernel, $\mathbb{K}$, is represented by a 5×5 matrix (yellow) as it moves onto different parts of the image. The position of the origin in $\mathbb{K}$ will be the position in the resulting image.

Depending on the type of filter applied, the values will change to highlight certain features in the image, like edges or corners. The following sections discuss two of the principal filtering approaches: smoothing and sharpening of image features. The difference lies in the properties of the kernels chosen to perform the convolution. To highlight the processes performed in each image, two levels of detail will be provided.

Smoothing filters

In this section, examples of smoothing filters are presented, as well as its application on relevant images for this work. The filters are presented in terms of the kernels that contain them.

Averaging kernel In this type of kernels, the values of the elements in $\mathbb{K}$ represents a weight or proportion, that is, the values are between $(0, 1)$ and the sum of all the values in the kernel add up to 1. With the previous explanation in mind, in the convolution $\mathbb{I} \circledast \mathbb{K} = \text{imfilt}(\mathbb{I}, \mathbb{K})$, every pixel value is replaced by the weighted average of the pixels in the n-neighbourhood.

The most straightforward implementation of this filter is what is known as a **box filter**, which has uniform weights assigned to all the pixels $\mathbb{K}(i, j) = 1/n^2$. The only parameter is the size of the filter, which at each point in the resulting matrix produces the mean of the intensity values contained within the scope of the filter. Another example of the weight distribution in the kernel is the **disk filter**, an averaging filter acting upon a circular area. The circle is or radius r and the size of the kernel is $2r + 1$. The filter highlights circular structures which match the size of the radius of the circle. The principle is the same as the one in the box filter, given the size of the filter, taking the central element $\mathbb{K}(r + 1, r + 1)$, all elements within the radius will contain a value higher than zero. As it is a discrete filter, the edges of the circle will not coincide with the finite grid of the kernel.

Gaussian filtering Gaussian filters are another type of smoothing filter in which the values follow a discrete 2D Gaussian of the size of the kernel. The parameters in the filter are zero mean (0) and variance (σ^2), as well as the size of the filter (n). As seen in Section A, the 2D shape of the Gaussian resembles a bell, where the width of it is determined by the variance. It is important to note that the Gaussian kernel must consider a value for n large enough for an adequate discretisation of $\mathbb{K}$.

Order statistic filtering. Most filtering is performed through convolution. As shown in Figure 3.6, the convolution can be interpreted as a moving *window* of the size of the kernel, which at every step, selects the pixels in the input image that will contribute to the new intensities certain position in the output; normally through the sum of the array multiplication of the elements in the kernel by the elements in the image.

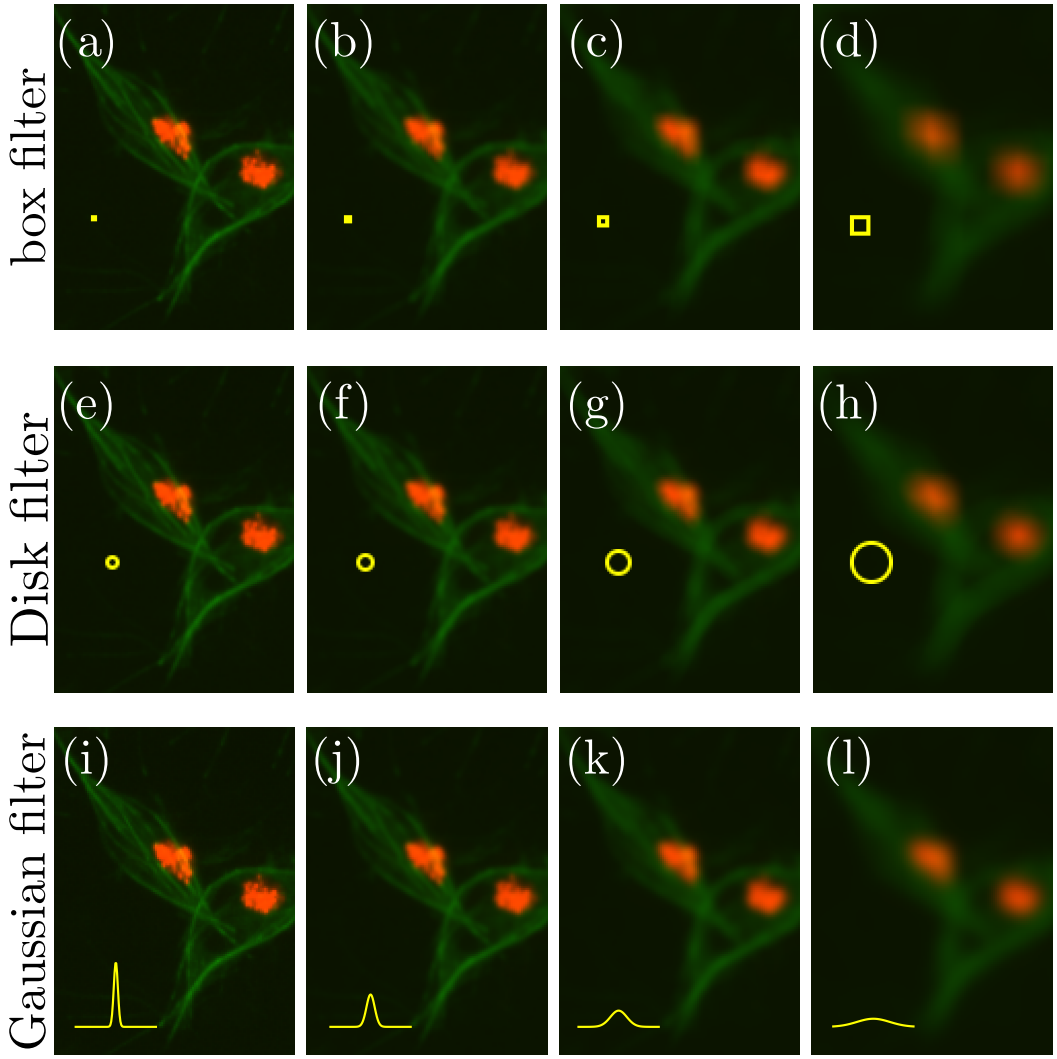


Figure 3.7 Representation of filters for the box (top row), disk (middle row) and Gaussian (bottom row) kernels with different parameters. For the box filters, (a) 2×2, (b) 4×4, (c) 8×8 and (d) 16×16; for the disk filters, the radii for each row are (e) $r = 2$, (f) $r = 3$, (g) $r = 4$ and (h) $r = 5$; for the Gaussian filters, the parameter varied was the variance σ^2. Detail of a frame is presented after applying a Gaussian filter with varying variance, σ^2. The outline of a 1D normal distribution, $\mathcal{N}(0, \sigma^2)$ was superimposed to the image for visualisation purposes. (i) $\sigma = 0.5$, (j) $\sigma = 1$, (k) $\sigma = 2$, (l) $\sigma = 4$.

Order statistic filtering, the same idea of a moving window is taken, but at every stage, the calculation of a percentile is performed in the pixels selected at a certain moment. The most common calculations are the minimum, maximum and median filters. Figure 3.8 displays the detail of three examples of order statistic filtering with various different sizes, observed in close detail.

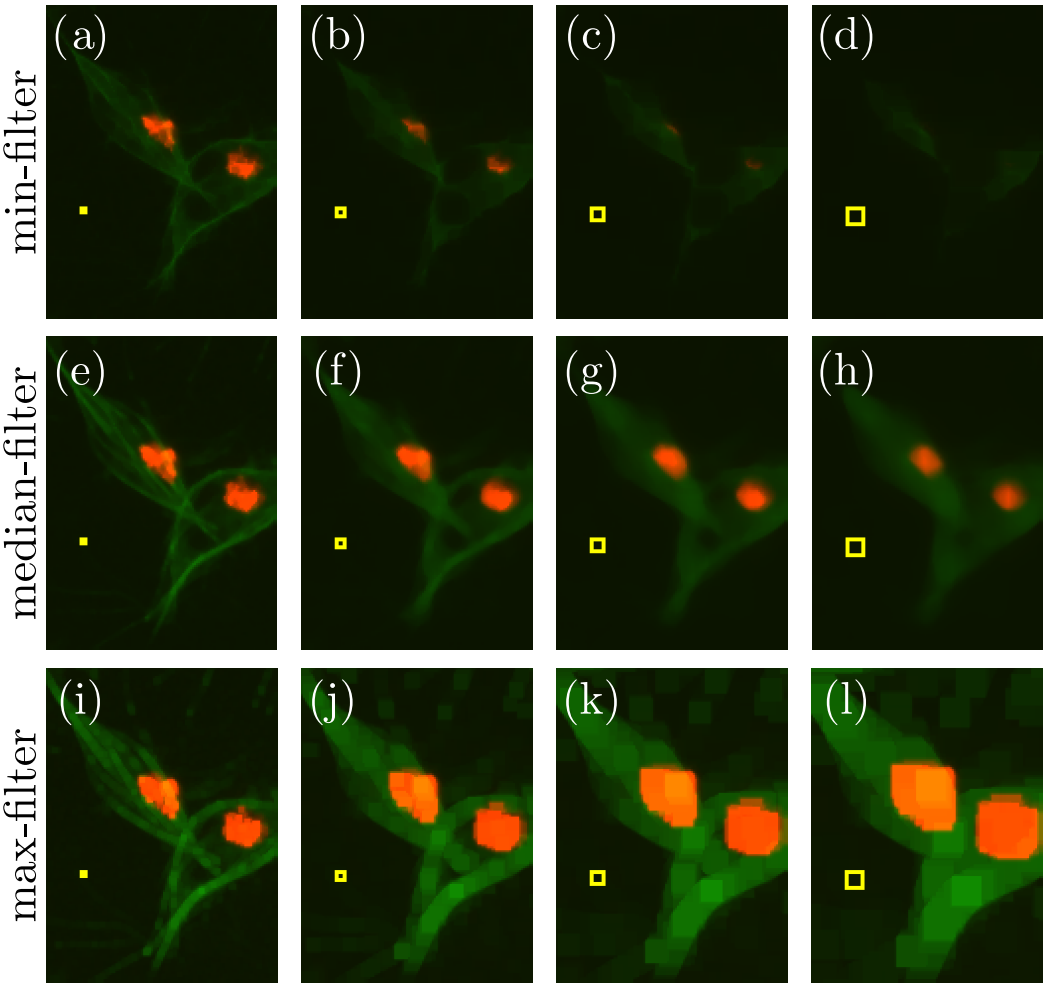


Figure 3.8 Representation of order different statistic filters with varying sizes. Detail of a frame is presented after applying three types of order statistic filters: minimum (top row: (a-d)), median (middle row: (e-h)) and maximum (bottom row: (i-l)). Each filter was tried using different sizes, and applied to each of the two channels to showcase the behaviour of each technique. A yellow box was superimposed to the image to aid the visualisation of the size of the filter. From the left: first column (a,e,i) 3×3; second column (b,f,j) 7×7; third column (c,g,k) 11×11; fourth column (d,h,l) 15×15

Discussion. In general, smoothing filters are useful if the data presents high frequency noise, and thus serve as *low pass filters*, in the context of signal analysis. Several filters were applied to the data exploring their individual parameters, order statistic filters appear to cause a deeper modification of the intensities in the image due to their nonlinear nature. From averaging filters, the difference between the different filters is not noticeable, refer to figure B.7 for a deeper insight.

Sharpening filters

To analyse a digital image, it is sometimes useful to locate certain geometric objects, such as lines which could correspond to edges and ridges in the image where certain key objects are located. Thus, highlighting transitions in intensity, with a sense of the magnitude of such transitions becomes key to image processing [67].

In calculus, the operator that describes the rate of change at a certain point is the derivative, or in multivariate calculus, the gradient. As images were constructed as discretisations of continuous functions of two variables, applying the numerical concepts of differentiation to the images would provide an insight to the characterisation of intensity transitions. On the other hand, as an image can be observed as a signal in the spatial domain, abrupt transitions could be linked to locations of a high frequency. In Figure 3.9, a combination of both ideas is represented, as this section describes the tools used to highlight transitions in images.

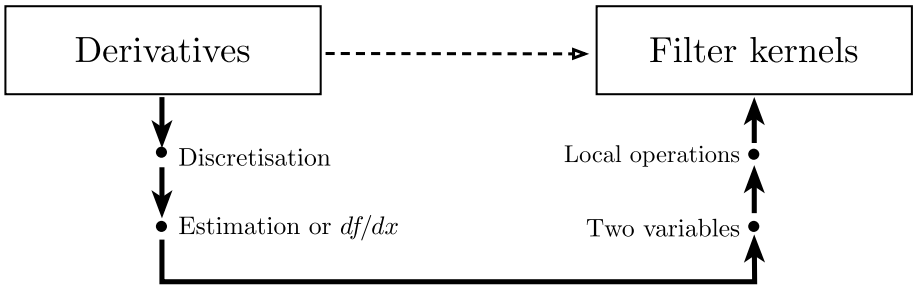


Figure 3.9 Relationship between derivatives and filters. In this section, the relationship between a derivative approximation for a 2D discrete function and a filter approach in which local operations performed for each point —or pixel— in the function.

Methods based on derivatives. In this section, methods based on derivatives will be described. The derivatives will be outlined both in terms of the gradient and the filter which can be used to perform the analysis. The **image gradient** can be obtained from an estimation of the derivative through central differences. In this work, the symbol used for partial derivatives applied to an image is $\partial_x \mathbb{I}(x, y)$, $\partial_y \mathbb{I}(x, y)$, will refer to the pointwise estimation of the derivative. While the symbol for gradient $\nabla_x \mathbb{I}$ will refer to the image which contains all the partial derivative calculations per position, *i.e.* $\nabla_x \mathbb{I} = (\partial_x \mathbb{I}(i, j))$. Consider a kernels given by the matrices $\mathbb{K}_x = (1, 0, 1)^T$ and $\mathbb{K}_y = (1, 0, 1)$. Centred around position (x, y), a convolution of certain filters in the image would produce at each pixel the approximation of the derivative per point.

The image gradient will consist of both images containing all the partial derivatives per point, and per direction: $\mathbb{G}_x = \nabla_x \mathbb{I}, \mathbb{G}_y = \nabla_y \mathbb{I}$, as well as the gradient magnitude $\mathbb{G} = |\nabla \mathbb{I}| = \sqrt{\nabla_x \mathbb{I}^2 + \nabla_y \mathbb{I}^2}$. Different calculations of the gradients can be applied, commonly, **Sobel**, **Prewitt** and **Roberts**. The difference lies in the kernels used. Table 3.2 shows the different kernels to calculate the gradients $\mathbb{G}_x, \mathbb{G}_y$ for an image. The **Lapalacian** filter is also presented, which provides the estimation of the Laplacian operator $\nabla^2 f = \partial^2_{xx} f + \partial^2_{yy} f$, a complete explanation on all sharpening filters is presented in appendix B.

Table 3.2 Table of kernels of common sharpening filters in different directions.

Methods based on derivatives			
Sobel	Prewitt	Roberts	Laplacian
$\mathbb{K}_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$	$\mathbb{K}_x = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$	$\mathbb{K}_{d1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\mathbb{K}_1 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$
$\mathbb{K}_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$	$\mathbb{K}_y = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$	$\mathbb{K}_{d2} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$	$\mathbb{K}_2 = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$

To showcase the different implementations of the derivatives, figure 3.10 includes the detail of a frame in the `MACROS1` dataset in which each channel has been processed with each of the filters.

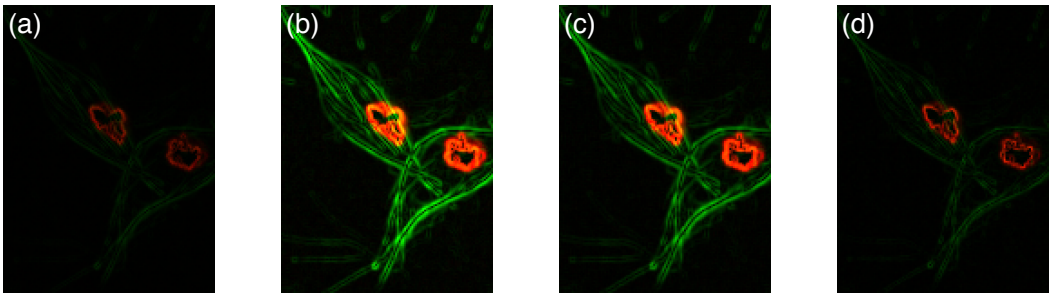


Figure 3.10 Representation of different sharpening filters. Detail of a frame is presented after applying the different sharpening filters analysed in this section, each one presenting the gradient's magnitude $\mathbb{G} = \sqrt{\mathbb{G}_x^2 + \mathbb{G}_y^2}$. (a) Central differences, (b) Sobel, (c) Prewitt and (d) Roberts.

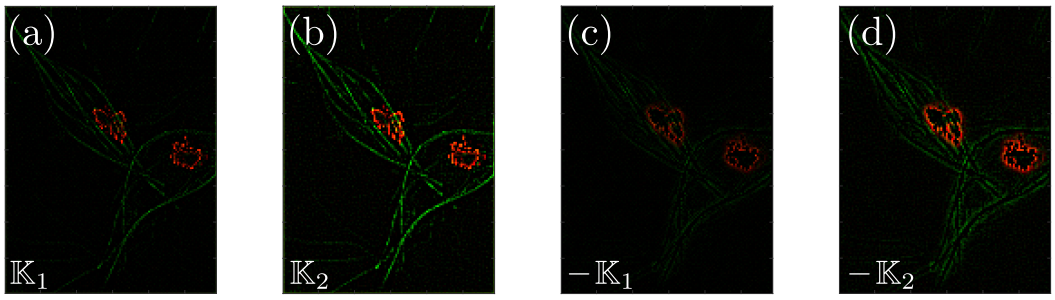


Figure 3.11 Representation of the different implementations of the Laplacian filter for $\mathbb{K}_1$ and $\mathbb{K}_2$. (a) $\mathbb{K}_1$, (b) $\mathbb{K}_2$, (c) $-\mathbb{K}_1$ and (d) $-\mathbb{K}_2$.

Applications of the Hessian matrix. Other implementations of second order derivatives include the Hessian matrix, which is a 2×2 matrix containing the second order derivatives with respect to x, y and the cross derivatives:

$$\mathbb{H}(f) = \begin{bmatrix} \partial^2_{xx} & \partial^2_{xy} \\ \partial^2_{yx} & \partial^2_{yy} \end{bmatrix} = \begin{bmatrix} \mathbb{G}_{xx} & \mathbb{G}_{xy} \\ \mathbb{G}_{xy} & \mathbb{G}_{yy} \end{bmatrix}.$$

As this matrix is produced per point, then four matrices will be calculated per image. This procedure is useful when some operations of the Hessian values are needed. For example, blob detection, done by the determinant of the Hessian $\text{DoH} = \mathbb{G}_{xx} \cdot * \mathbb{G}_{yy} - \mathbb{G}_{xy} \cdot \wedge 2$, or in **ridge detection**, where the eigenvalues of the Hessian per pixel are found by the following equation. An example frame is shown in figure 3.12.

$$\lambda_{\max}(\mathbb{H}) = \mathbb{G}_{xx} + \mathbb{G}_{yy} \pm \sqrt{\mathbb{G}_{xx} \cdot \wedge 2 + \mathbb{G}_{yy} \cdot \wedge 2 - 2\mathbb{G}_{xx}\mathbb{G}_{yy} + 4\mathbb{G}_{xy} \cdot \wedge 2}.$$

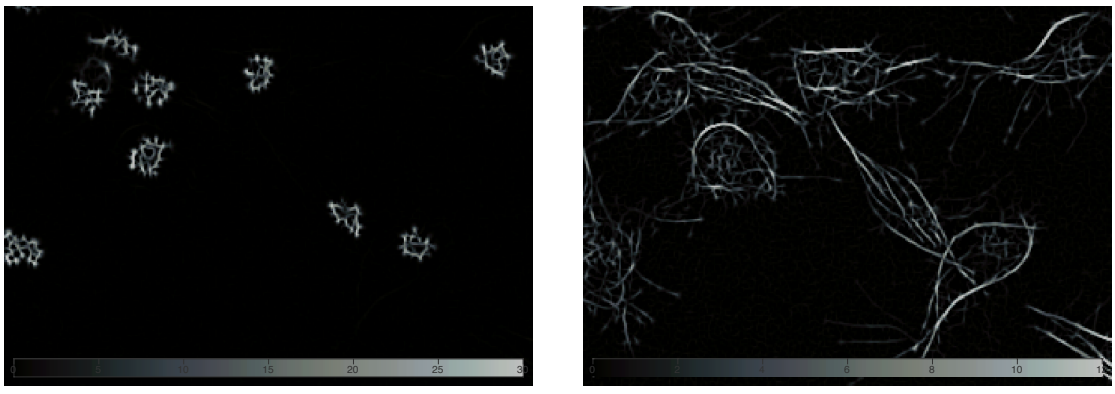


Figure 3.12 Example of the minimum eigenvalue of the Hessian matrix per pixel. The difference between edges and can be observed.

Edge detection An important problem in image analysis is the detection of contours of objects or edges. Note that filtering highlights edges and points of abrupt changes, however edge detection refers to the creation of a binary image where positive values are located where changes are more abrupt. In particular, the Canny [79] edge detection algorithm has been a widely referenced image analysis technique with almost 30 thousand citations since its publication in 1986[1]. As it is relevant to this work, an overview of the algorithm is provided and some examples of its implementations, while varying its key parameters. The overview of the algorithm is described in Algorithm 3 in Appendix B.

The two main advantages of the algorithm are (i) the ability to modify the Gaussian kernels, allowing different edges to be visible and (ii) the non-maximum suppression which allows to select the more relevant ones. The technique would be better classified as a segmentation of the edges in the image, as the output is a binary image with two levels, $I = 0, 1$. It is included in this section to provide context of the application of sharpening filters and because the term Segmentation, in this work, refers to the distinction of cells from other cells and from the background.

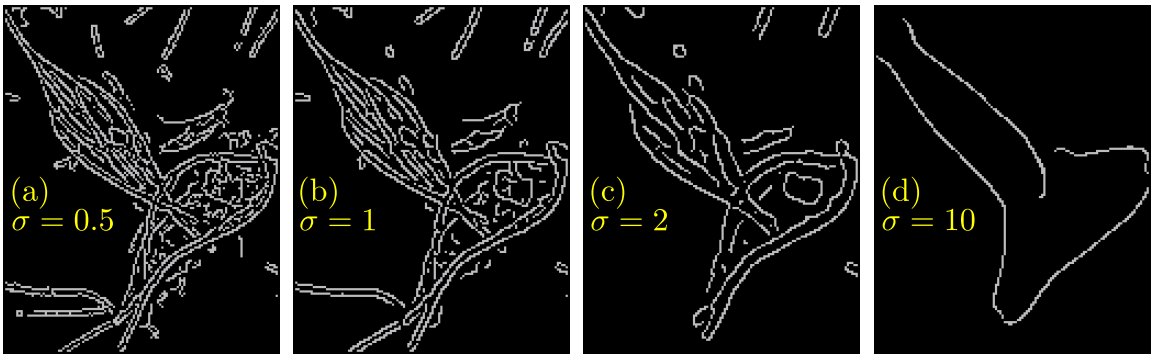


Figure 3.13 Detail of output of Canny algorithm on macrophages data. Different values of σ^2 are presented to showcase the output of the algorithm. Notice that the size of the variance σ is inversely proportional to the level of detail being analysed by the algorithm.

Discussion. As observed in figures 3.10 and 3.11, and more importantly in Figure 3.13, the variable nature of the images of macrophages prevent the filtering to produce clear edges.

[1]Consulted in Google Scholar, August 2018

Image segmentation by intensity thresholding

Segmentation is the process of classifying pixels into one of two categories: foreground, or objects of interest, and background. In a way, segmentation provides context to an image selecting the regions which —given an application— are important. In particular, cell segmentation is a widely studied problem which has produced considerable amount of research output, like the works by Maška et al. [68] and Ulman et al. [64].

Considering the definitions in Section 2.1, intensity levels on an image are referred to be a finite set of L levels, $\{1, 2, 3, \ldots, L\}$. Thus, performing a segmentation by intensity thresholding implies finding a value k which separates the intensity levels into groups: $\{1, 2, 3, \ldots, k-1\}$ and $\{k, k+1, k+2, \ldots, L\}$. Let $\mathcal{C}_R$ be the set of intensities in which the intensities of an image are categorised, where R refers to the region in which the image will be segmented, and can be either a subscript or a number depending on the context. It is important to remember that the segmentation output of a method will be a binary image which takes the value 1 at the detections of foreground and 0 at the background. As it is relevant to this work, the concept of **hysteresis thresholding** [80], mentioned in Section 3.1, is a technique for image segmentation that uses two levels to segment an image, a higher one that determines what will be part of the foreground and a lower one that will distinguish what will be considered as background. A comprehensive explanation and examples can be found in the Appendix B. In the following sections, the description of different techniques to automatically select the threshold from the image intensities in the image. Two main approaches are compared: the global threshold, which uses a single level throughout the image and the adaptive threshold which selects a threshold locally.

Otsu's threshold

Otsu [81] developed an algorithm for differentiating between classes, *i.e.* finding a threshold, by maximising the quotient of *between* and *within* variances of the classes. In simple terms, the algorithm would select a threshold that would distinguish the classes as much as possible, while at the same time would make the intensity levels within the class as alike as possible, one advantage is that this procedure can simply be extended to more than two classes. The derivation of the algorithm is presented for a single threshold as the extension to multiple thresholds is straightforward.

Given the levels of intensity found in an image $I = \{1, 2, 3, \cdots, L\}$, the Otsu method finds a level of intensity k, $1 \leq k \leq L$, that maximises the quotient σ_B^2/σ_W^2,

where σ_B^2 is the variance between the two classes (foreground and background) and σ_W^2 is the variance within the classes. A full derivation of the equations can be found in B.

Figure 3.14 displays the detail of a macrophages frame and the regions created after intensities get classified after applying different Otsu thresholds.

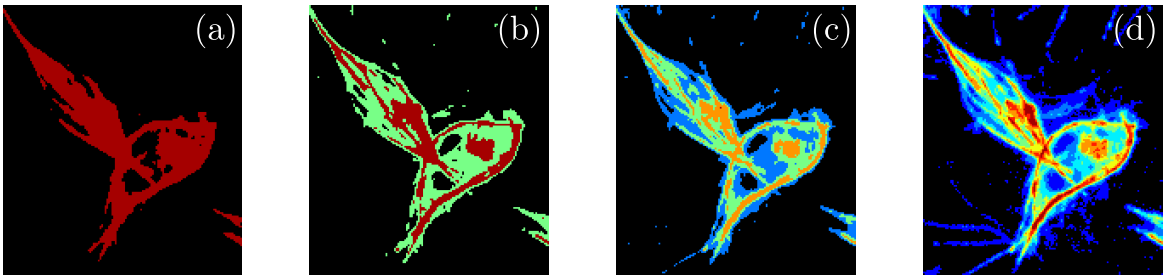


Figure 3.14 Detail of two macrophages after applying different Otsu thresholds. The Otsu algorithm was applied to the same section of a frame. Different number of thresholds were applied, producing different number of classes $\mathcal{C}_i$. (a-d) show the results of applying 1,2, 4 and 8 thresholds selected by the Otsu method.

Otsu's method, while very fast, is very sensitive to outliers on the data. One example can be studied on the ISBI datasets, with the Fluo-N2DH-GOWT1 dataset. It can be observed in Figure 3.15(b) how the difference between the most and least intense is too high for the algorithm to segment both cells. Also, it can be noted in 3.15(b) that even taking a robust approach like a hysteresis threshold is not enough for cells of this kind.

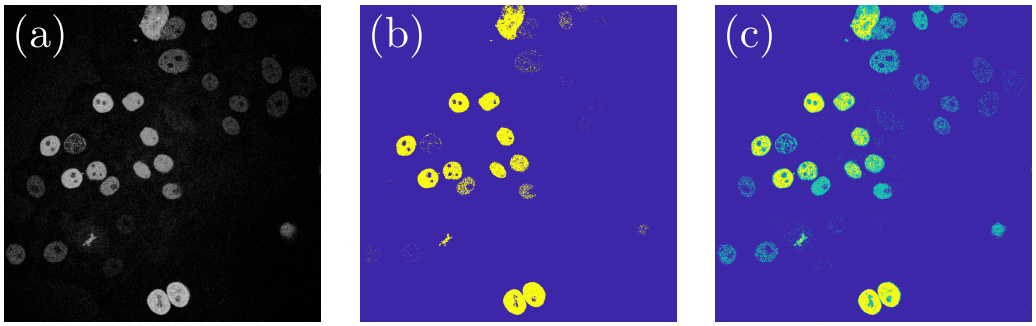


Figure 3.15 Limitations of the Otsu algorithm in frame from the ISBI dataset. (a) Represents a frame from the Fluo-N2DH-GOWT1 dataset, recalling section 2.2. One and two thresholds were applied to the image, and the resulting classifications of the pixels are presented in (b) and (c), respectively. It is important to notice that the variability of intensities in the foreground is too large, and the algorithm is incapable of segmenting some of the darker cells.

Adaptive thresholding

Variations of the thresholding technique are presented in the works by Pappas [82] and Bradley and Roth [83]. In the work by Pappas [82], the segmentation achieved was different from the global ones, as the threshold level chosen is dependent on the local pixels at given positions. Instead of selecting thresholds that take into consideration the statistical properties of the entire image, the method starts with global estimates of the levels and adapts them to the local characteristics of each region.

The concept of an adaptive threshold becomes relevant to this problem, as the objects of interest in fluorescent images can vary considerably, and a single threshold might not be enough, even a robust one like a hysteresis threshold, as seen in Figure 3.15. In addition, it is possible that the images suffer from shading due to a variety of reasons [84]. The work by Pappas [82] can be seen as a generalisation of the k-means algorithm [85] that include spatial constraints and account for local variations of intensity. The algorithm works through the estimation of the regions of an image (segmentation) and the parameters in an *a posteriori* density function that computes the probability of the distribution of regions (x) given the observed image (y). A thorough explanation of the process is available in Appendix B.

On the other hand, the work by Bradley and Roth [83] (B&R) presents an adaptive threshold methodology that takes into account spatial variations in illumination. It achieves its purpose by selecting a threshold value based on local mean intensity in the neighbourhood of each pixel. A neighbourhood of 1/8th of the size of the image is used around each pixel. Each pixel gets assigned a different threshold value in which each pixel is compared against. A parameter called *sensitivity* is included, which allows for some control from the user into the threshold taken by the algorithm. Figure 3.16 shows a frame in the macrophages dataset which has undergone the two adaptive threshold methods overviewed in this section.

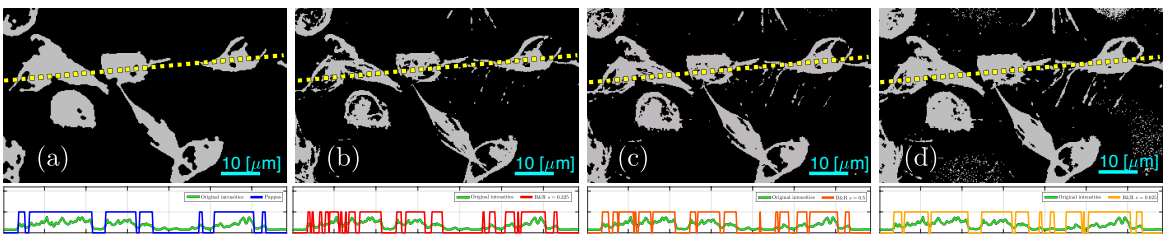


Figure 3.16 Comparison of different adaptive threshold techniques. The same representative frame from the dataset as well as the image profile marked in the same spot (yellow, dotted). (a) Shows the adaptive threshold as described by Pappas [82]. (b-d) Represent the results B&R [83] algorithm with the sensitivity set to 0.325, 0.5 and 0.625 respectively.

Postprocessing

Postprocessing is the set of operations performed on binary images which aid the segmentation to reduce noise or over-detection. In the context of segmentation of cells, some of the operations can aid in reducing noise, smoothing the edges of the shapes and help give a more robust segmentation. Measurements can be performed on the binary images, like area and orientation.

In this section, some techniques of morphology are overviewed. Morphology in this work refers to *mathematical morphology*, the branch of mathematics that deals with the representation and description of region shapes in a space [67]. In Section 3.1, the construction of filter kernels and the notion of convolution was explored in order to modify the intensities by assigning an operation performed on a neighbourhood of a pixel. To extend the notion into binary images and sets, some operations must be defined in the context of sets, and then translated into binary arrays to define each of the operations.

Structuring elements and notation. Sets are represented by connected regions in an image with the same value. As mentioned in the classic work by Gonzalez and Woods [67, Chapter 10], a *structuring element* is a small set of subimage used to probe an image under study for properties of interest. Such objects can be thought of in abstract as sets with a centre, or reference point, or as a binary kernel, represented by the symbol $\mathbb{S}$.

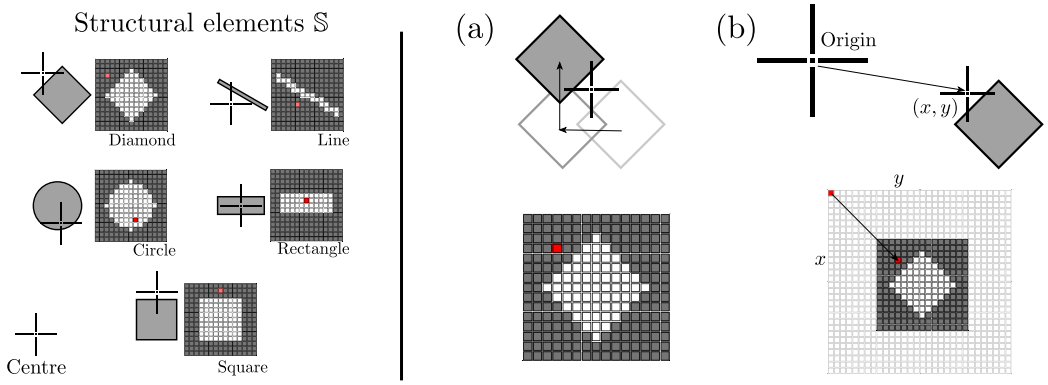


Figure 3.17 Representation of different structural elements in abstract terms and as a binary kernel $\mathbb{S}$. The diagram shows the abstract representation of a structuring element as a set with a particular shape. The notion of a centre refers to a point of reference in which the element can pivot and from which it can move in a plane. (a) Shows the operation of reflection while (b) shows translation. Notice that if the structural element is symmetric, and the centre is in the midpoint of the element, then $\hat{\mathbb{S}} = \mathbb{S}$.

Morphological operations

In most cases, morphological operations are performed similarly to convolution, as seen in Figure 3.6, as a structuring element will be translated along a binary image, performing set operations as it moves. In this section, the following operations will be defined: (i) erosion, (ii) dilation, (iii) opening and closing; as well as some algorithms like (iv) boundary extraction and (v) hole filling. For the coming definitions, let $\mathbb{I}$ be an image and $\mathbb{S}$ a structuring element of size $2n+1$, with a centre at position $(n+1, n+1)$. For simplicity, assume that $\mathbb{I}$ contains only one set —or detected object— in it. The operations will result in new sets, described by the elements in it. In terms of images, the positions that belong to a particular set will have a value of 1 in the resulting image.

Table 3.3 Description of main morphological operations and equations in the context mathematical morphology. Let $\mathbb{I}$ be a binary image and $\mathbb{S}$ a structuring element of size $2n+1$.

Name	Mathematical expression	Description
Erosion	$\mathbb{I} \ominus \mathbb{S} = \{(x,y) \vert \mathbb{S}_{x,y} \subseteq \mathbb{I}\}$	Reduce the area of present regions in $\mathbb{I}$
Dilation	$\mathbb{I} \oplus \mathbb{S} = \left\{(x,y) \vert \hat{\mathbb{S}}_{x,y} \cap \mathbb{I} \neq \emptyset\right\}$	Expands the area of present regions in $\mathbb{I}$.
Opening	$\mathbb{I} \bigcirc \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}$	Erosion followed by a dilation.
Closing	$\mathbb{I} \odot \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}$	Dilation followed by erosion.

Figure 3.18 represents the application of the different morphological operations of a structural element $\mathbb{S}$ into a set with disjoint regions $\mathbb{A}$. The structural element present in the picture will be translated throughout the plane containing the set $\mathbb{A}$, and the output produced will follow the equations in Table 3.3.

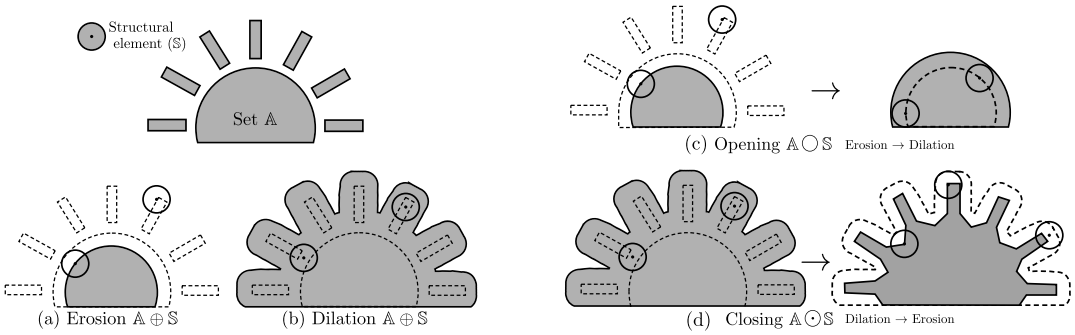


Figure 3.18 Diagram of morphological operations. The set $\mathbb{A}$, is represented by different disjoint regions, while the structural element, $\mathbb{S}$, is represented as a circle, with its centre in its midpoint. The outputs of the different operations can be observed: (a) erosion, (b) dilation (c) opening and (d) closing.

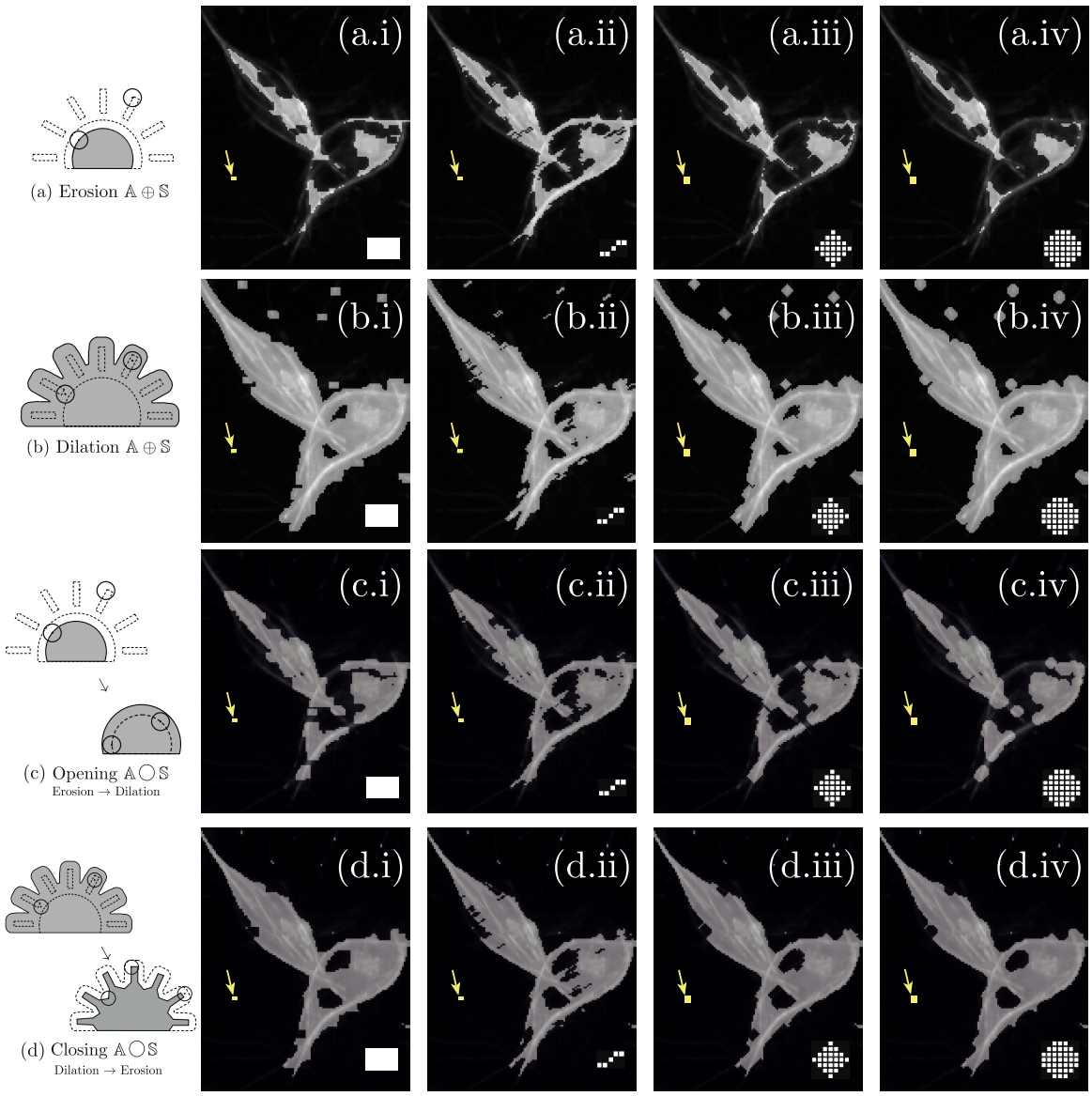


Figure 3.19 Example of morphological operations outputs when applying different structural elements. All images present a binary image overlapped on a section of a frame in the macrophages dataset. The actual size of each structural element is highlighted in yellow, and marked with arrows. The shape of the structural element kernel $\mathbb{S}$, is shown amplified at the lower right corner of each image. Rows represent the operations used: (a) erosion, (b) dilation (c) opening and (d) closing; while the columns represent the elements: (i) rectangle, (ii) line, (iii) diamond and (iv) disk.

Discussion. Combinations of techniques like in morphological opening, could be useful, as the erosion could reduce granular noise and the subsequent dilation would grow the area of the detected object and close certain gaps between disjoint sections.

Feature extraction in images of macrophages

Another useful way to analyse images, involves extraction of certain characteristics in the image called features. Feature extraction is common in problems of texture analysis, where detection of objects is based on characterisation of regions in an image by the variations of intensity values of an image, sometimes modelled through its statistical properties. Texture analysis provides a quantitative description of qualities —such as *smooth* or *rough*— in image regions.

Features are extracted by filtering an image with a number of specific kernels to produce different related images, $\{\mathbb{I}_q\}_q$, all images contain information extracted locally by the different kernels applied. Thus, per pixel, a vector of length q holds local information extracted from the local characteristics of the image. Each pixel could then be grouped with similar pixels through the similarity of the feature vector extracted. As a continuation of the data exploration, two techniques are considered to the analysis, Gabor filters [86] and SIFT/SURF techniques, [87, 88].

Gabor filters

Gabor filters [86] refer to kernels which —as seen in the filtering section of this work— assign to each pixel local information about it. The kernels have a specific orientation and frequency, or wavelength. A common application for texture classification [86] involves using a bank of filters with varying frequencies and orientations to extract a large vector which contains different information on each pixel, and then using a statistical technique such as k-means [85] to cluster them.

Implementation. Figure 3.20 show an example of a standard Gabor filter bank, in which the variation of wavelength and orientations can be observed. The orientations shown are increments of 30 degrees from 0 to 150. The wavelengths were chosen in increasing powers of two, starting at $\lambda_0 = 4/\sqrt{2}$ and up to the hypotenuse length of the input image, in this case $\lambda_{\max} = \sqrt{N_r^2 + N_c^2}$, as stated in the work by Jain and Farrokhnia [89]. This would produce six wavelengths and six orientations, for a total of $q = 36$ kernels. In Figure 3.20, only a fraction of such kernels are shown for visualisation purposes with different sizes. Note that the sizes displayed in the figure were chosen arbitrarily, and are not to scale.

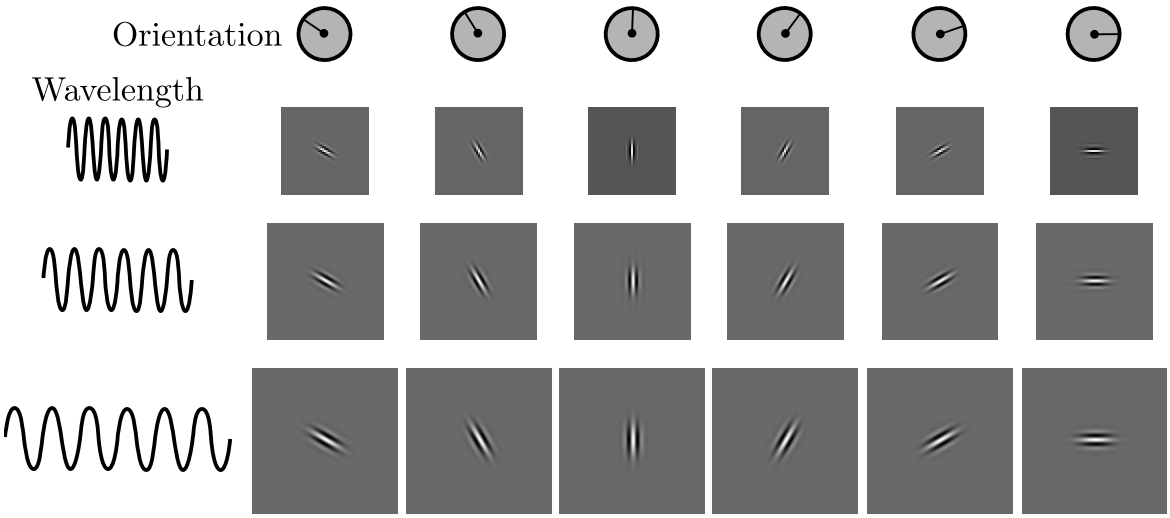


Figure 3.20 Example of a Gabor filter bank for feature extraction. Notice the change in orientation of the kernels and the change in size and resolution as the wavelengths increase.

After obtaining the filter bank, each filter is convolved with the frame to produce $q = 36$ images, $\{\mathbb{I}_q\}_q$. Figure 3.21 shows the different instances of $\mathbb{I}_q$, after convolving with the Gabor filters in the bank. Notice how each kernel, due to the wavelength and orientation highlights different areas of the original frame.

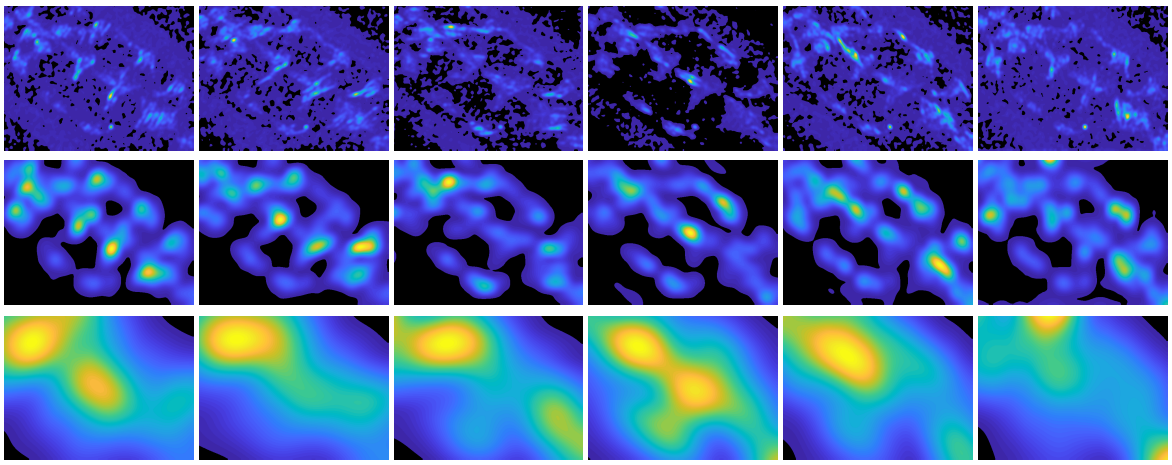


Figure 3.21 Convolution output of the different Gabor kernels on the green channel. Each Gabor filter kernel shown in Figure 3.20 was convolved with the green channel of the test frame. Notice that as the size of the filter increases, due to the wavelength increase, the output of the convolution highlights different areas of the figure of different scales.

Feature vectors are then formed by aligning pixels at same positions in each of the generated images, $\mathbb{I}_q$. Therefore, each pixel in the original image has a vector in $\mathbb{R}^q$ associated to it and the local information from it. K-means [85] is then used to group

the different vectors into one of two groups, foreground and background. Figure 3.22 represents the output of the experiment.

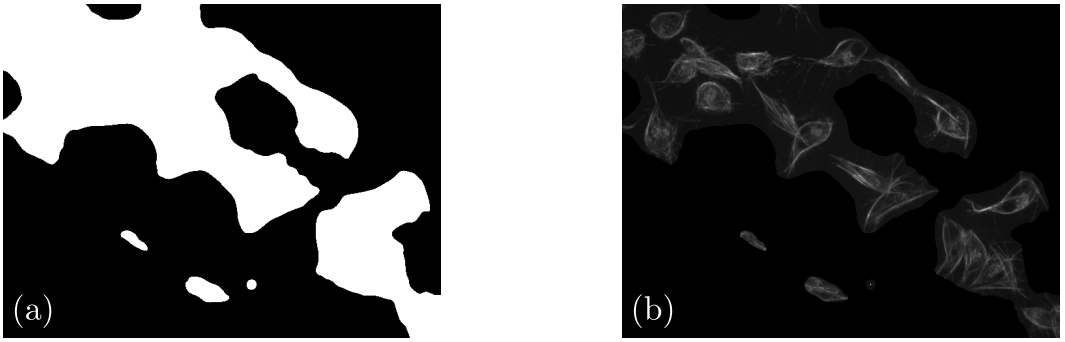


Figure 3.22 Foreground and background detected using Gabor features. (a) Represents the binary image of foreground (white) and background (black). (b) Shows the original frame, with the background suppressed for visualisation purposes. Notice that the technique was able to identify areas where cells exist, but the segmentation is not done, as not even single cells were detected.

Scale-Invariant Feature Transform and Speeded-Up Robust Features

In an image, a local feature refers to a pattern of specific structure like a point, edge or a small image patch. The **Scale-Invariant Feature Transform** was introduced by Lowe [87] as a novel class of local feature detection, which were impervious to transformations such as translation, rotation or even illumination changes. On the other hand, the **Speeded-Up Robust Features** (SURF) [88, 90] were proposed by Bay et al. [90] as a method to obtain a scale and rotation-invariant feature detector that could outperform SIFT.

In this section, a brief description followed by an experimental study on the algorithms' ability to follow single cells in consecutive frames are presented. Furthermore, in Appendix B, a thorough description of SIFT and its differences to SURF are described and an illustration is provided to highlight the algorithms' advantages.

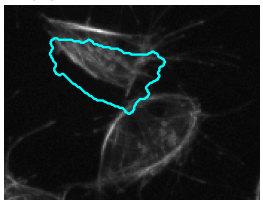
The SIFT and SURF algorithms are centred in finding keypoints in the context of their orientation. The keypoints can be understood as a point on an image chosen by some criterion, in the case of SIFT, local maxima and minima on specific functions of the input image. For it to be robust, it is necessary to guarantee that the image is considered at distinct scales, which is achieved by convolving the input image with Gaussian kernels of varying standard deviation σ. **The SURF algorithm** uses the same idea as the SIFT algorithm, however it applies key changes to the implementation

of its parts in order to improve performance and running time. While SIFT calculates image pyramids and then computes the DoG, SURF creates a stack of images without downsampling which combined with the use of integral images, filters the stack using a box filter approximation of the second order derivatives of Gaussians, which allows for computation of rectangular filters in near constant time [91].

Implementation. A normal implementation which exploits the features created by SURF, involves two images, a reference one which contains the object of interest and a secondary image which contains the object in a particular *scene*. The SURF algorithm would be run on both images and two sets of features would be found. Since SURF detects features regardless of transformations, the same features would be found in both images, and the feature vectors could be compared between them, matching points in the original image and the *scene*.

For the implementation using the `MACROS1` dataset, a similar approach was made. Given the temporal resolution of the cells, the changes from frame to frame only vary slightly. Therefore, the features detected by the SURF algorithm would be anticipated to be similar around the same cells in both consecutive frames. This means the representation of each cell per frame would be similar enough in consecutive, or even close frames to produce the appropriate features to provide an area. Assuming a previous segmentation has been made, based on intensity thresholding techniques, the concept of a *clump* was presented in Section 2.1, and in Figure 2.17. Considering a scenario in which one cell is segmented on its own, and in the following frame, the segmentation has produced a clump. The experiment is based on detecting the keypoints from the SURF algorithm on both images, taking the first frame as the reference and the secondary as the target. In figure 3.23, a detail of the cell of interest is shown with the outline detected by the segmentation; both consecutive frames are shown, and the *clump* in the target frame is outlined. However, on visual inspection both frames appear similar.

(a) Reference cell



(b) Target cell in clump

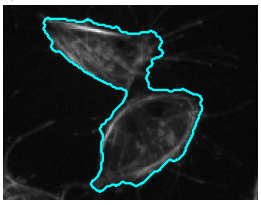


Figure 3.23 Cell in two consecutive frames for experimentation of SURF in `MACROS1` dataset. (a) Represents the original frame, in which the cell of interest was completely detected by the segmentation algorithm. (b) represents the same cell in the consecutive frame. The segmentation consists of a Gaussian smoothening, followed hysteresis thresholding where the levels were chosen with the Otsu [81] algorithm. Finally, morphological operations were done as postprocessing, overviewed in section 3.1.

First, SURF features were detected on the entire frame to see the potential of the algorithm with this data. The default parameters of feature threshold and number of scales was used, and then parameters which would produce more keypoints was searched for empirically, the detail of the parameters and the matched points from the reference to the target frames can be found in figure 3.24.

Finally, a cutout of the cell of interest was used as the reference image, to investigate whether the information detected in the cutout frame would produce enough keypoints which would be matched in the entire target image. In figure 3.25, the matched keypoints of such experiment can be seen. The most important parameter modified is the number of scale levels per octave computed in the algorithm. In MATLAB® the default parameter is of 4 scale levels per octave, while the empirically chosen parameter was of 9 scales. The default number of octaves computed is three. The results in figure 3.25 show only two points being matched between the reference and target.

Discussion. The objective of this implementation is to explore the possibility to create an affine transformation of the matched points in the reference to the target frame. The transformation would then be applied to the boundary of the correctly segmented cell in the reference frame and use it as the segmentation of the same cell in the target frame, solving the ambiguity of the *clump*. In the case of the experiment described in figure 3.25, only two points in the reference were matched to the target frame, and while the points correspond correctly, only two points are not enough to produce a transformation. Hence an approach like this would not be suitable for tracking the cells after overlapping.

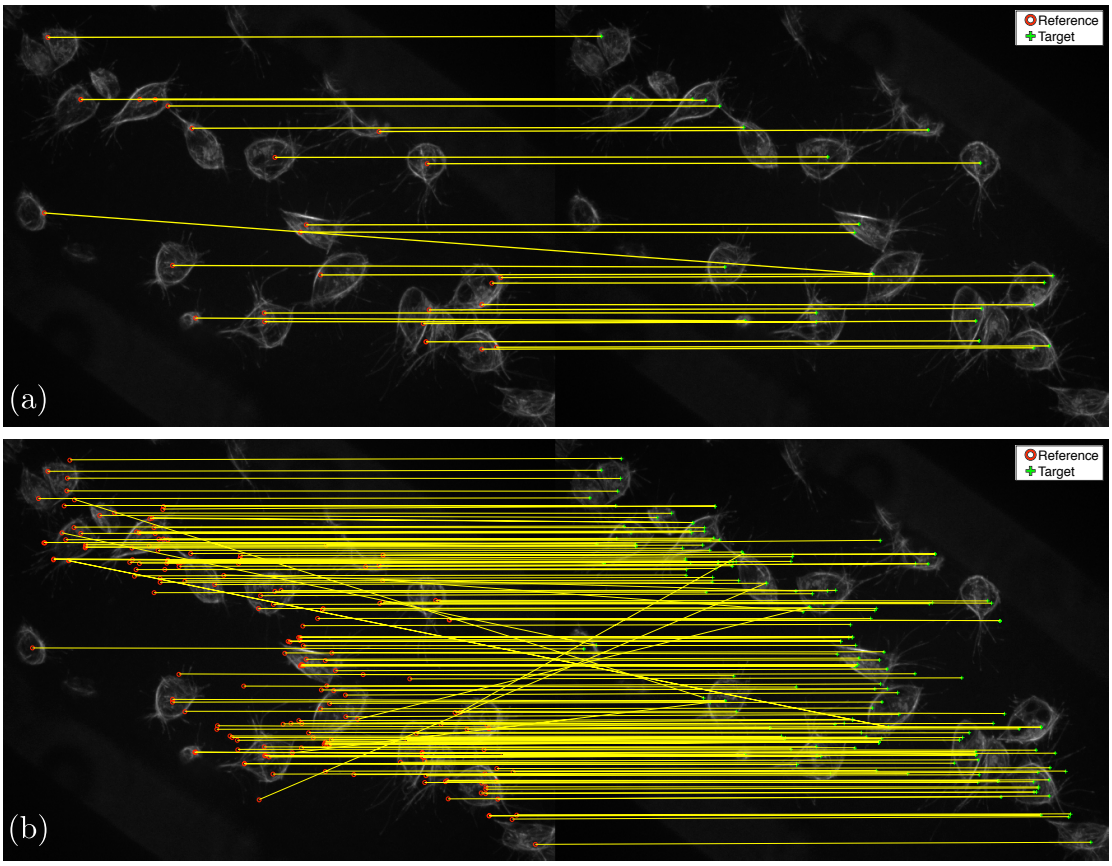


Figure 3.24 Experimentation of SURF algorithm in macrophages data using different parameters of threshold and number of scales. The implementation was done in MATLAB® with the default parameters of a threshold of 1000 and 4 scales (a) and compared with the points matched with a threshold of 50 and 9 scales (b). The principal difference observed is in the number of matched points. Notice that in both experiments, some points are matched to different locations.

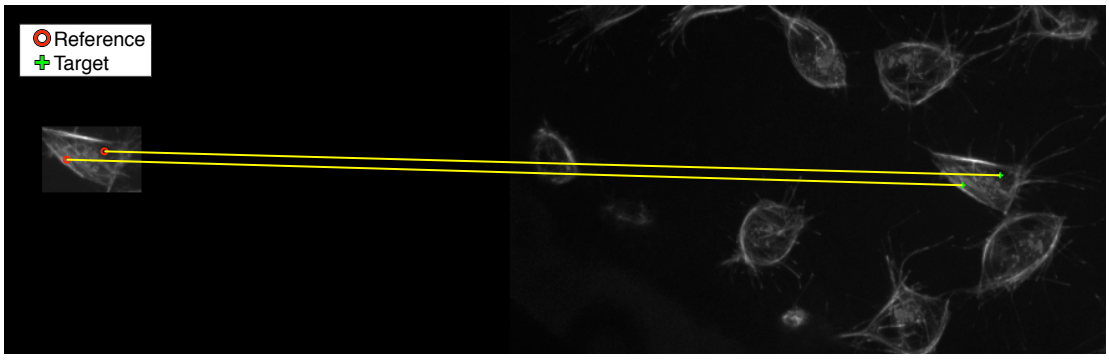


Figure 3.25 Experimentation of SURF algorithm from the region surrounding the reference cell to the target frame. Only two points were matched from the reference cutout frame to the target.

3.2 Segmentation techniques of overlapping objects

This section shows some techniques explored during the development of this project to analyse single frames of overlapping macrophages. Segmentation by intensity thresholding is central in this work. However, due to its simplicity, it introduces ambiguity in the segmentation when used with the macrophages datasets, due to the interacting cells, which cause overlapping. Thus, more sophisticated approaches need to be explored to resolve the ambiguity. The methods alongside the implementations are described, and a brief discussion on each topic is presented in terms of the lessons learned by implementing each one of them. First, a straightforward approach using Voronoi Tesselations was developed. Such naïve approach could provide insights of the worst-case scenario of a segmentation that tries to solve overlapping of the objects. Then, a statistical approach is discussed, through Active Shape Models [92], which allowed to understand some of the difficulties inherent of the macrophages dataset. Curve evolution methods were also explored, which provided some interesting ideas which will be exploited in upcoming chapters. Finally, a brief overview of *machine learning* techniques is presented, specifically neural networks.

Voronoi Tesselations

Voronoi Tesselations (or Voronoi diagrams) [93, 94] involve a polynomial partition of a plane on a set of points scattered through it, where each point is associated with the section of the plane nearest to. Let $\mathcal{P} = \{\mathbf{x}_i\}_{i=1}^n$ a set of n points in a plane $\mathbf{x}_i \in \mathbb{R}^2$, which will be referred to as *sites*. Two concepts are provided to describe the relationships of the points in the plane to the *sites* that generate the polygonal partition, these are the **dominance of power** of a point $\mathbf{x}_i$ over another $\mathbf{x}_j$ and the **region of a site**; both are sets that describe areas in the plane. This technique studies the image as a set $\mathbb{U}$, where all positions are present.

Definition 1 (Dominance of power) *Let* $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{P}$ *be two distinct* sites. *The dominance of power* $\mathrm{dom}(i,j)$ *of* $\mathbf{x}_i$ *over* $\mathbf{x}_j$ *is the set of points in the plane which are closer to* $\mathbf{x}_i$ *than to* $\mathbf{x}_j$

$$\mathrm{dom}(i,j) = \left\{ \mathbf{x} \in \mathbb{R}^2 \middle| \, \mathrm{dist}(\mathbf{x}_i, \mathbf{x}) \leq \mathrm{dist}(\mathbf{x}_j, \mathbf{x}) \right\},$$

where $\mathrm{dist}(\cdot,\cdot)$ *is the Euclidean distance.*

Definition 2 (Region of a site) *Let* $\mathbf{x}_i \in \mathcal{P}$*, the site region,* $\text{reg}(i)$*, is defined as the intersection for all sites* $\mathbf{x}_j \in \mathcal{P} - \{\mathbf{x}_i\}$ *of the dominance of power* $\text{dom}(i,j)$

$$\text{reg}(i) = \bigcap_{j \neq i} \text{dom}(i,j)$$

Figure 3.26 shows the explanation and relationship between the two concepts previously presented. The Voronoi partition will be formed by the union of all site regions for all points in $\mathcal{P}$.

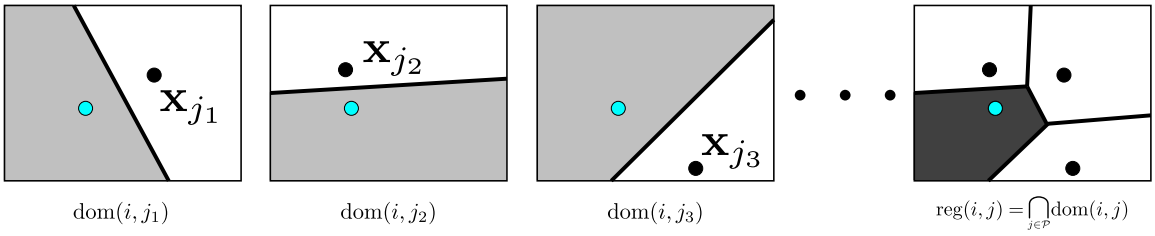


Figure 3.26 Illustration of a Voronoi Partition. From left to right, the calculation of three cases of $\text{dom}(i, j_x)$, relative to a given point $\mathbf{x}_i$ (cyan) can be observed. To the far most right, the site region $\text{reg}(i)$ is represented. It is easy to understand that the union of all the site regions in a plane, $\mathbb{U}$, corresponds to the Voronoi partition of it.

Implementation with overlapping macrophages The implementation incorporated a hysteresis thresholding segmentation of the both the green and red channels. As seen in Section 3.1, the green channel would produce clumps, illustrated in figure 2.17. However, the segmentation of the red channel would produce a robust segmentation of the nuclei. Using the centroids of the nuclei as the collection of points $\mathcal{P}$, the clumps could be partitioned based on the Voronoi diagram. Notice that some of the nuclei are not visible in the images due to imaging problems. Therefore, when two cells are found in a single partition, an artificial nucleus is incorporated into the image, and the partition is computed again. This process solves the problems, but introduces some extra nuclei in the images. Figure 3.27 displays two cases of the artificial nuclei, one correct and one incorrect. Figure 3.28 shows a comparison of each cell segmented with the Voronoi partition method described in this section vs. the ground truth in a frame. Notice the colour map follows black for TN, red for TP, blue for FP and yellow for FN.

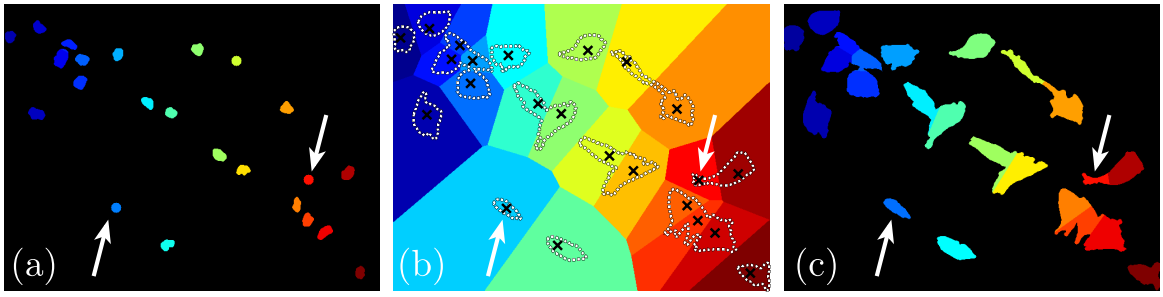


Figure 3.27 Naïve segmentation using Voronoi Tesselations. In this example, the test frame can be observed. First, the segmented nuclei are represented in (a), each in different colour. (b) Shows the partitioned space with the centroid of the nuclei marked (black ×); for visualisation purposes, the outline of the detected clumps and cells is marked (white - -) (c) shows the output of the Voronoi segmentation. Notice the two arrows, in (a), they correspond to artificial nuclei incorporated by the software as the original partition would create more than one detections in the green channel per single nucleus. The arrows are repeated in (b,c) to highlight the outcome of the algorithm's incorporation.

Discussion on the Voronoi implementation The Voronoi partition based on the centroids of the nuclei in the image was developed as a way to represent a naïve approach to separating clumps into single cells. The algorithm was never meant to solve the problem, but to provide a baseline in which future developments would be compared. Other methodologies, sophisticated or simple, would require better results as the ones highlighted here.

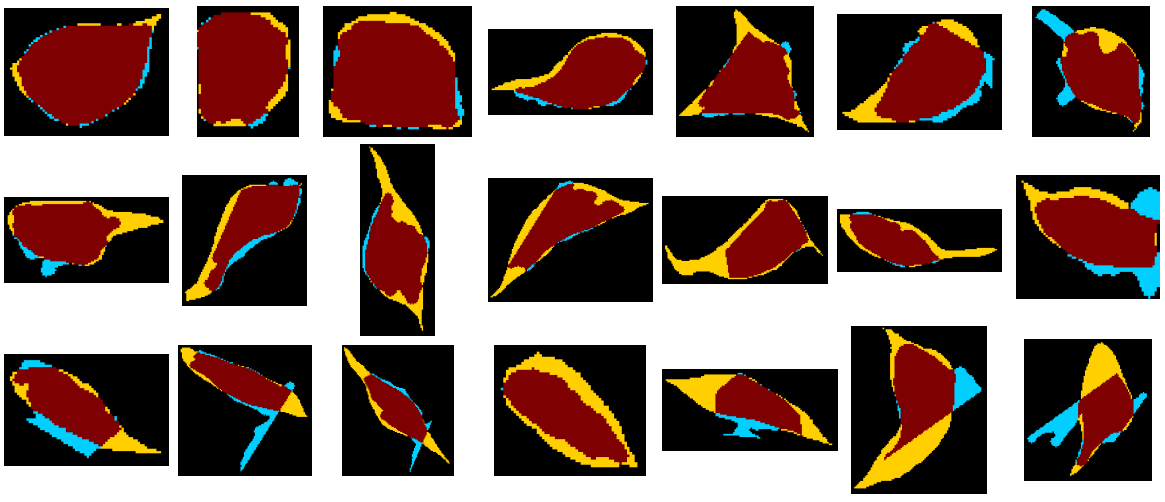


Figure 3.28 Results from implementation of Voronoi partition in overlapping macrophages. See full explanation in text.

Active Shape Models

Active Shape Models (ASM) [92] is a technique in which known objects are located in images through a model of their physical shape. In the model, the natural variability of the object class is captured by the algorithm to find examples of structures. Although the model has been extended [95, 96], the basics are still referenced in the literature. In this work, ASM was implemented, with the original intention of making use of the rigidity innate to the Point Distribution Model generated by the algorithm. The implementation details and results obtained are explained in this section. The aim is building a model which describes the typical shape and variability of a class of object. Each object is characterised by a set of ordered landmark points, which the input to the model consists of a set of individuals (instances) of the objects.

Let $\mathbf{x}_i \in \mathbb{R}^{n\times 2}$ be a vector describing the n landmark points of an instance i in an object. To compare equivalent points from different shapes, they must be aligned with respect to a set of axes. Alignment is done by a rotation and scaling of the landmark points. Therefore, let $R(s,\theta)[\mathbf{x}]$ be a rotation by θ degrees and a scaling in s. A translation vector, $t_j = (t_{x,j}, t_{y,j})$ is used to centre the landmark points. Figure 3.29 summarises the approach taken by this method.

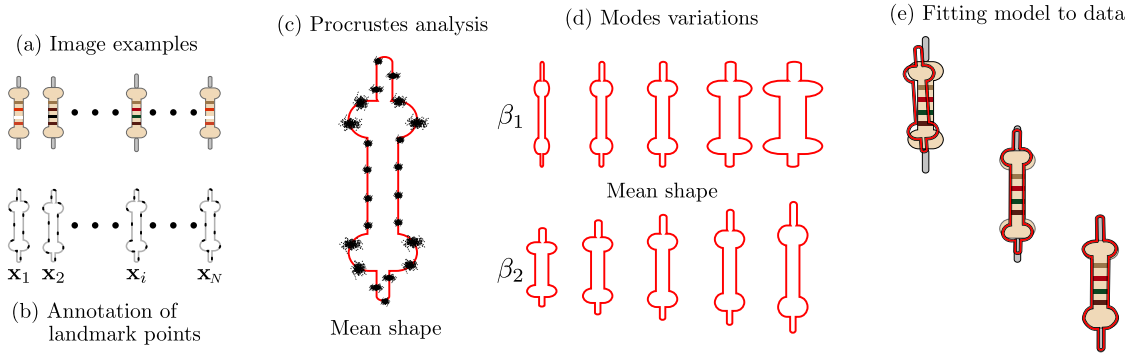


Figure 3.29 Graphical representation of the ASM algorithm. From the examples of the images of the object to be modelled (a), the points $\mathbf{x}_i$ are built by manually annotating landmark points (b). The Procrustes analysis is performed on the resulting points and a mean shape is built (c), the variation of the captured points can be seen around each landmark point. The PCA model is built, finding the main modes in which the figure can be reshaped (β), two examples are shown in (d). Finally, using the mode, the model is fit to the data by altering the mean shape via the modes, depending on the shape change needed (e).

Variations for overlapping cell segmentations As seen in Chapter 2, the microtubule structures vary in shapes and sizes considerably. The ASM algorithm requires

as input a set of instances of the objects characterised by landmark points, however, the images described in Section 2.2, do not show distinct points which consistently remain in the cells' shapes. The single point, which is consistent along all shapes and all points, is the nucleus in each cell. Therefore, to implement the algorithm in this instance, landmark points were generated automatically using the position of the nucleus and taking equidistant points along the boundary of the generated ground truth of the macrophages.

In the work by Plissiti and Nikou [96], a similar problem was found since the cells in the pap smears analysed also lacked clear landmark points to train the algorithm. The way it was overcome in such a work was by implementing a model of the outline of the cells. Unfortunately, this was not viable as the shapes shown by the macrophages was so variable (recall figure 2.15). The order given to the points was based on the polarity of the cells, *i.e.* the vector formed by the centroids of the (green) microtubules with respect to the centroid of the (red) nuclei. The rest of the positions would be equally spaced and starting at the closest point in the direction of said vector and the outline of the green channel, Figure 3.30. Per shape, the output of this selection of 32 *pseudo*-landmark points produces a matrix of size 32×2, called $\mathbb{X}_i$. The set of all matrices for all points n_{tp} is allocated in a 3D matrix of size $32 \times 2 \times n_{tp}$, with the symbol $\mathbb{X}$: this arrangement of the data allows for a simpler programming.

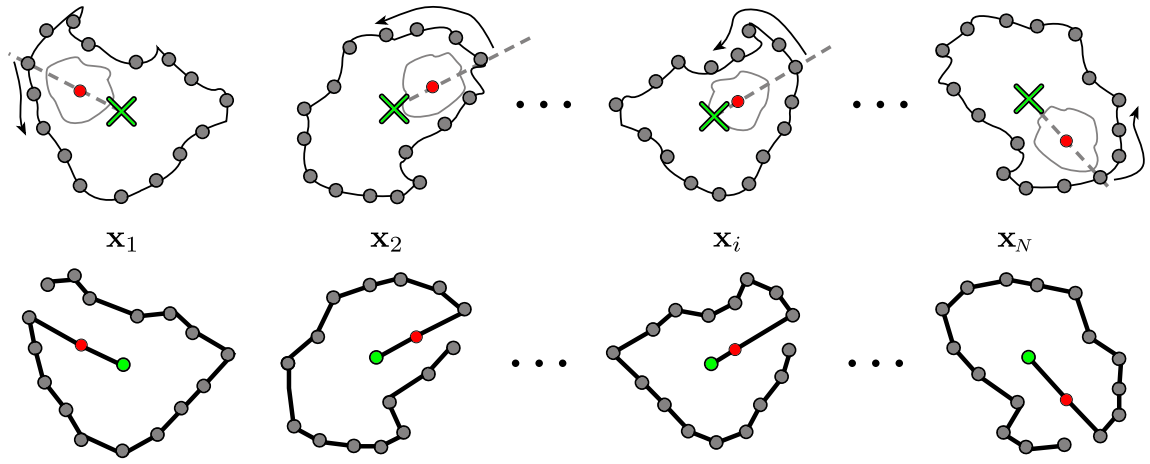


Figure 3.30 Landmark points determination in different instances of macrophages. Four different macrophages cells are shown. The first two points correspond to the centroids of the cell in the green channel ($\mathbf{x}_1$), and nuclei ($\mathbf{x}_2$). The following points fall along the boundary, being $\mathbf{x}_3$ the point which intersects the boundary of the shape in the green channel and the line that passes through points $\mathbf{x}_1$ and $\mathbf{x}_2$.

Details of the implementation. Ten frames were annotated manually using the code described in section 2.4. Macrophages in which the nucleus is not visible, were discarded from the analysis. Retrieving each of the training matrices $\mathbb{X}_i$ is done as described in Figure 3.31. Then, following the methodology in [92], a Procrustes analysis [97] is made to align the shapes to a single set of axes. The mean shape is computed, $\mathbb{X}_\mu$, which allows for the shapes to be centred. Finally, the data is rearranged by stacking the coordinates of each point into a matrix of size $64 \times n_{tp}$, then the covariance matrix $\mathbb{D}$ is calculated from the resulting columns. PCA is then performed over matrix $\mathbb{D}$, and the resulting model for a shape is given by,

$$\mathbb{X}_\beta = \mathbb{X}_\mu + \beta\mathbb{P}, \tag{3.1}$$

where $\mathbb{P}$ is the model which results from the PCA processing. Figure 3.31 shows a diagram of the representation of the data during the implementation of this technique. On the other hand, figure 3.32 shows the different modes created for each shape.

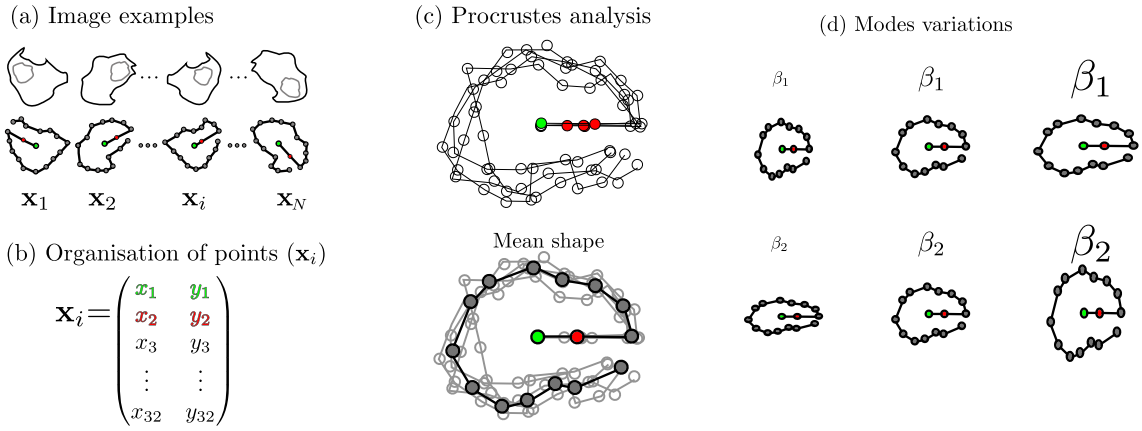


Figure 3.31 Implementation of the ASM methodology in macrophages data. (a) Extraction of points from boundary of green channel and centroids of green and red detection. (b) Organisation of points into 32×2 matrix $\mathbb{X}_i$. (c) Procrustes analysis over all $\mathbb{X}_i$ shapes and calculation of mean shape. After PCA analysis, (d) shows the modes of variation.

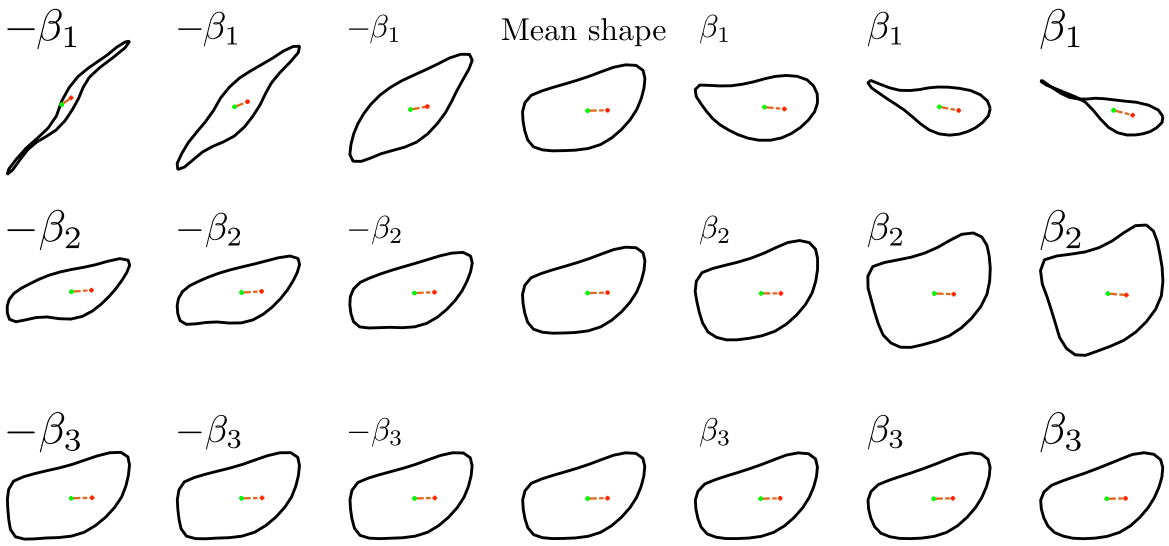


Figure 3.32 Results of the ASM model implementation. The model in equation (3.1) was trained for ten images where two cells were segmented. The different modes are shown as they vary from the mean shape $\mathbb{X}_\mu$.

Discussion on the ASM implementation. While the training of the method seemed to work successfully, various problems arose during the implementation of the fitting of the model to the data, which rendered it an unsuitable method to deal with overlapping segmentation. The wide variation in shapes and sizes of the cells would imply that matrix $\mathbb{P}$ to have many different columns —up to eight in some analyses— which defeats the purpose of explaining the variability of the shapes in a simpler model.

Furthermore, in order to adapt a shape to an image which had not been trained presented more challenges. First of all, the intensities of the overlapping cells are extremely confusing for a method to adapt to, and no initial shape would be suitable. Even in the example presented in 3.32, where only two similar cells were used for training, the implementation of the *step* that updates the shape proved difficult, and the initial shapes would not update quickly. The explanation could be that the choice of landmark points including the centroids could have caused problems.

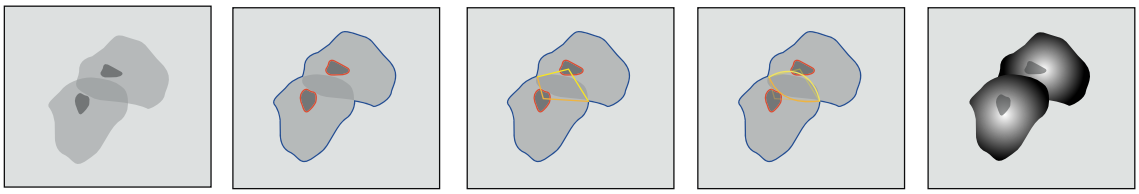
Level set methods

Level set methods are a type of curve evolution algorithms, techniques used for segmentation in which the outline of a detected object is considered as the contour at

the level of a 3D function. The mathematical theory of partial differential equations in image analysis is introduced in Appendix D, from the classical reference [98].

Lu et al. [99] developed a joint level set method technique to solve overlapping cases of pap smears. The core of the method involves level sets operating simultaneously, one per cell that has been detected as overlapped. The method requires two segmentations as input: one corresponding to the nuclei of the cells, and another one of the cells' cytoplasm, which contains the overlapping *clumps*. The algorithm detects the intersections of the boundaries of each cell and initialises the level set function per detected cell.

The images of pap smears only contain a single greyscale channel, and thus part of the methodology involves the detection of the nuclei present in each frame as well as the detection of each clump. Once these were found, an *a priori* mask —which assumes the shape of each cytoplasm is ellipsoidal— is generated. Each of the level set functions is applied and evolved to find the best fit to the boundary of the cytoplasm.



(a) Original (b) Clump and nuclei detection (c) Association of clump to nuclei (d) Extrapolation of boundary (e) Shape prior

Figure 3.33 Overview of the framework proposed by Lu et al. [99]. The first steps, (a-c), involve the separate detection of the clumps and the nuclei in the cells. Notice that the data in Lu et al. [99]'s work are simple greyscale images, therefore the task of separating both objects of interest becomes much more complicated. (d) Represents the extrapolation of the boundary from the clump, which includes the detection of the junctions where the boundaries meet and an estimation of the boundary based on ellipses. The overall shapes of the pap smear cells allow for such a simplification. Finally, (e) shows the creation of the shape prior, which will allow for the level sets to adapt to the outputs.

Implementation on images of overlapping macrophages.

After verifying the results obtained from the data presented in [99], the method was implemented on frames of dataset `MACROS1` with available ground truth. The detection of *clumps* and nuclei is straightforward for the macrophages data, therefore, the first steps of the function were not implemented. A user can select the initial stage of the

analysis as the data relevant to each stage is stored in the separate folders, which are accessed sequentially and produce specific output that points the program to a certain stage in the algorithm. Figure 3.34 represents the processing segmentation performed on the two channels and used as input to the LSM. The segmentation of the channels followed the same steps described before in the Voronoi partition, a hysteresis threshold as used and artificial nuclei were incorporated where it was needed. Both channels were input into the algorithm at the right step in the LSM framework (figure 3.33(b)).

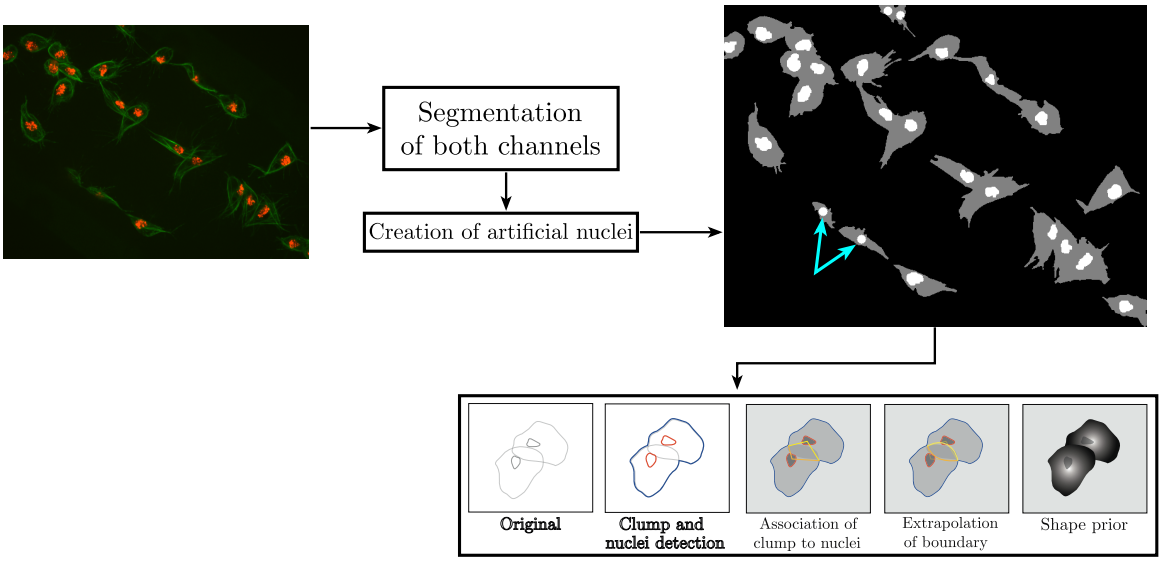


Figure 3.34 Description of input presented to joint level set method. Similar to the Voronoi implementation, the segmentation of the two channels included a hysteresis threshold, with the levels chosen with Otsu [81]'s method. Artificial nuclei were incorporated where it was needed, marked with cyan arrows.

Discussion on the Level Set method implementation. In their work, Lu et al. [99] reported promising results segmenting some cases of overlapping pap smears. However, as seen in figure 3.35, the method was not able to produce satisfactory results with the macrophages data. The main reason involves the extrapolation of the boundaries of the participating clumps. The original algorithm estimates the positions of the points where the boundaries intersect my measuring the distance of the nuclei to each other. It also assumes the shape of the cell to be circular or ellipsoidal. Neither assumptions are useful for the macrophages as the the cells vary widely in size and shape. However, an important discovery for this project was made after the exploration of this technique. The estimation of the location of the **boundaries intersections** was the main problem in this implementation and the results obtained were found

to be relevant to this work. Potentially better results might be produced if a more robust approach in detecting the intersection of two overlapping cells. This notion was explored and later implemented in Chapter 4.

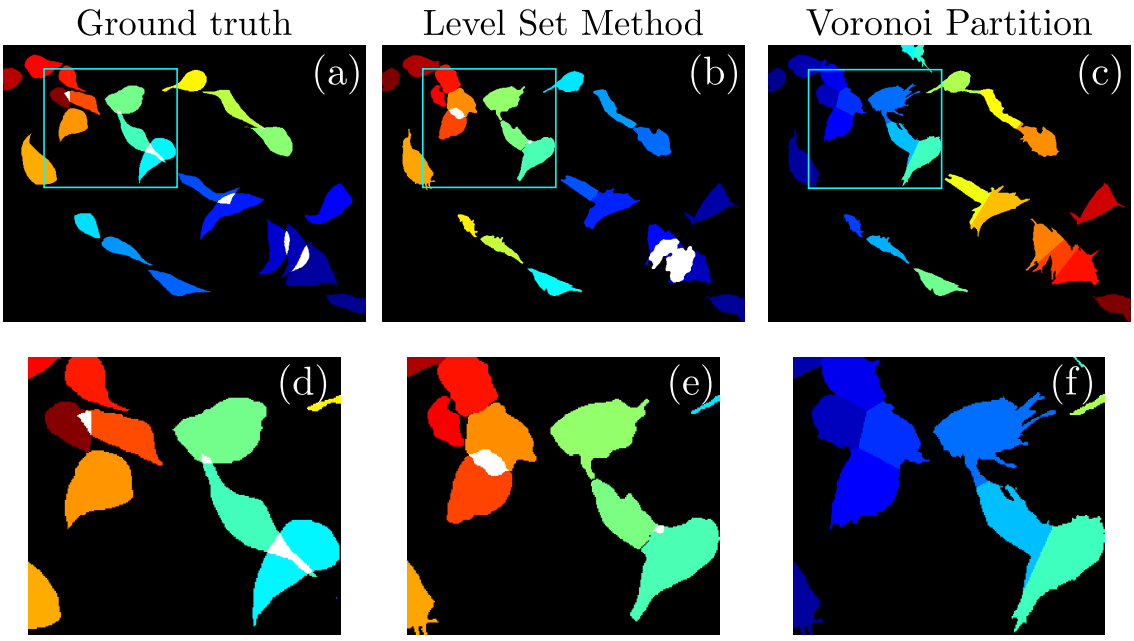


Figure 3.35 Comparison of the joint Level Set Method with the Ground Truth and a Voronoi partition. The top row (a-c) shows the full segmented frame with a box highlighting some interesting cases. In the bottom row (d-f), the detail of the box is shown. In all cases, different cells are coloured and regions where there is an overlapping have been drawn in white. (a) The ground truth of the green channel of the frame is presented as described in Section 2.4. (b) Shows the output of the joint LSM by Lu et al. [99] and (c) the output of a Voronoi partition as described in Section 3.2.

3.3 Tracking

Tracking, in the context of this work, is defined as linking a single object as it migrates in time from several segmentations in images from consecutive time frames. This work considers Multiple Particle Tracking, which means that it is possible that more than one object will have to be linked between every pair of frames. The input for the tracking process, will be the output of a segmentation process in which each individual object will be assigned a unique label to be identified. Several different techniques link, with varying degrees of precision, the labelled objects. There are methods where the tracking is coupled with the segmentation, which means that the methods look for specific features of the objects to correlate one frame to another. An example is the

work by Collins and Carr [100] who have proposed that tracking and segmentation could be studied as a single complex problem. However, this work will focus on an approach decoupled from the segmentation; the algorithms assume that at least spatial coordinates are provided. The notion of having standalone segmentation and tracking techniques paired together makes sense in the context of the project. Given that the objective is to provide a software framework for the analysis of tracks of macrophages, then having the option to pair and combine the various techniques could offer more possibilities for even more applications.

In this section, the keyhole tracking algorithm [101, 59] is described, alongside the PhagoSight framework [102], which utilises such tracking. A thorough description of the algorithm and framework is provided in appendix B.

Nomenclature and notation. Regarding the links between objects in different frames: regarding an object in frame t, the object in frame $t-1$ that is linked to it will be referred to as *parent*, and the object linked in frame $t+1$ will be referred to as *child*. Tracks will be presented as a 3D visualisation where the coordinated axes correspond to rows, columns and time frames or (x, y, t). Finally, each object tracked will include an identifier r, and the set of points $\{(x_t, y_t)\}_t$ will include the symbol $\mathcal{T}_r$. Figure 3.36 represents a diagram of the result of a tracking experiment.

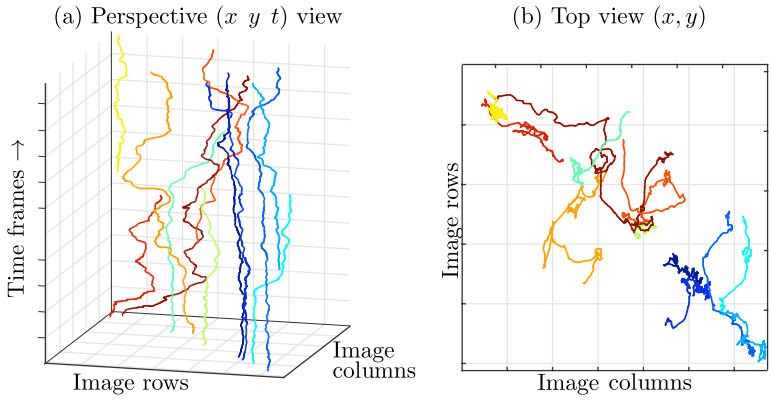


Figure 3.36 Illustration of tracking experiments. Each tracking experiment will produce several tracks, each corresponding to an object moving in time. (a) Represents a 3D visualisation where the horizontal plane span the number of rows and columns in an image, the bottom represents the first frame. Also notice the different starting points of each track and different lengths. (b) Represents a top view of the same tracks, where the temporal component is lost in exchange for a more precise understanding of the spatial movements created buy the tracks. In general, longer tracks are considered to be more consistent. Shorter tracks are usually cases where a more thorough analysis is required.

PhagoSight and the Keyhole tracking algorithm

To explore the tracking on different datasets described in 2.2, the PhagoSight framework is briefly introduced. The full description of the software can be found in Appendix C. PhagoSight [102] is a software framework originally designed for the segmentation and tracking of neutrophils and macrophages observed in zebrafish. The segmentation module involves a hysteresis threshold based on the single threshold Otsu algorithm. The tracking module of PhagoSight is the **keyhole tracking algorithm** [17, 59, 102], and consists of an estimation of the position at the child frame following the velocity from the parent frame, it creates two regions of probability which appear like an old keyhole, thus the name of the algorithm. The output of the PhagoSight framework consists of a large table containing information from each of the tracks detected, given a particular segmentation. Table 3.4 shows the output information from the PhagoSight framework.

Table 3.4 Track information retrieved from PhagoSight framework. The information utilised for the analysis is highlighted: position provides the coordinates x_t, y_t, z_t of the cell in the image per frame t, time frame is the temporal position in the dataset, t, label in binary image refers to the label assigned to the cell at segmentation and final track label corresponds to the track identification, $\mathcal{T}$.

Position (x_t, y_t, z_t)	Distance to closest	Time frame
ID	Parent cell	Child cell
Velocity	Volume	Label in binary image
Keyhole	Track ID	Final track label
Bounding box information	Volume to surface	Sphericity
Neighbours at different distance brackets	Distance to disappear	Distance to appear

It is important to notice that PhagoSight does not account for overlapping objects or cellular processes such as mitosis, thus the one of the two cells generated would be regarded as a new cell and a new track identifier would be assigned to it. Finally, the PhagoSight framework includes a set of tools to amend and manually edit the tracks, either in code or via a GUI. In the development of this PhD project, the PhagoSight software was implemented in two separate instances: with the ISBI datasets, described in section 2.2 and with *Artemias*, Section 2.2. In the following sections, a brief description of such implementations is presented, as a way to explore the track outputs which could be generated with PhagoSight, but mainly, with the tracking algorithm.

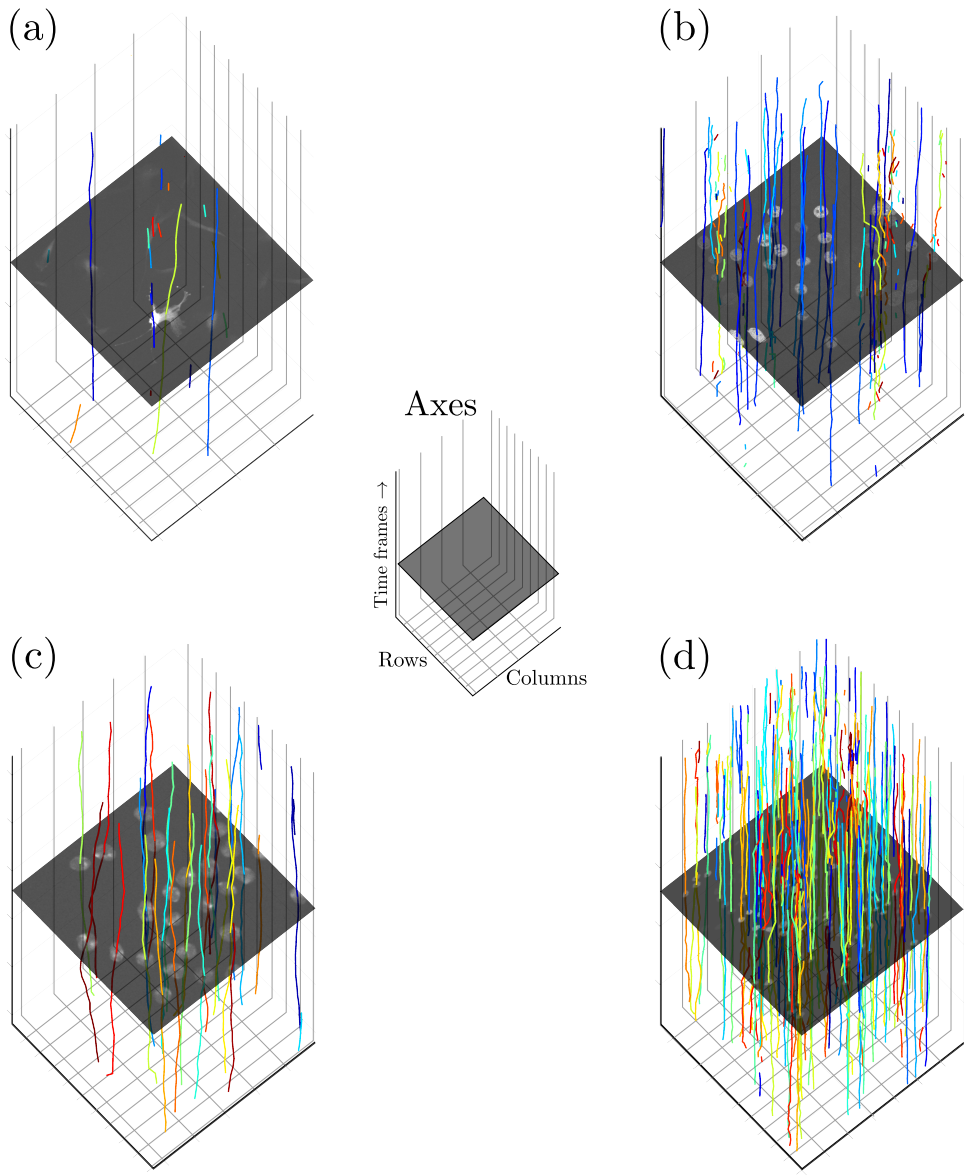


Figure 3.37 Example of implementation of PhagoSight on different ISBI data. A fraction of the tracks detected of four datasets is presented: (a) C2DL-MSC, (b) N2DH-GOWT1, (c) N2DH-SIM and (d) N2DL-HeLa. The colour of the track represents the relative speed of the track. In all results, (a-d), the x and y axes represent the rows and columns of the image and the time frames are read from the bottom of each plot to the top. A guide for the axes is presented in the middle of the picture. (a) **C2DL-MSC** shows few consistent tracks and some very short tracks indicating to some possible segmentation errors. (b) **N2DH-GOWT1** shows slower tracks, in shades of blue, to be much more consistent than faster and shorter ones. These are usually at the edges, which could indicate that the imaging technique used did not observe the cells at the edges of the image. (c) **N2DH-SIM** shows mostly consistent tracks. (d) **N2DL-HeLa** shows a large number of tracks, if looked closely, some smaller tracks can be observed. In this dataset, cases of mitosis can be observed, spawning new tracks.

Implementation for the *Artemia franciscana*. For the *Artemia* dataset, the preprocessing of the frames included an earlier stage that consisted of two earlier stages: manually selecting a region of interest in the data, and removing the mean image of all the frames. Figure 3.38 shows the preprocessing of the images. Notice the change in the image profile selected, and how it would affect the attempt to threshold the images.

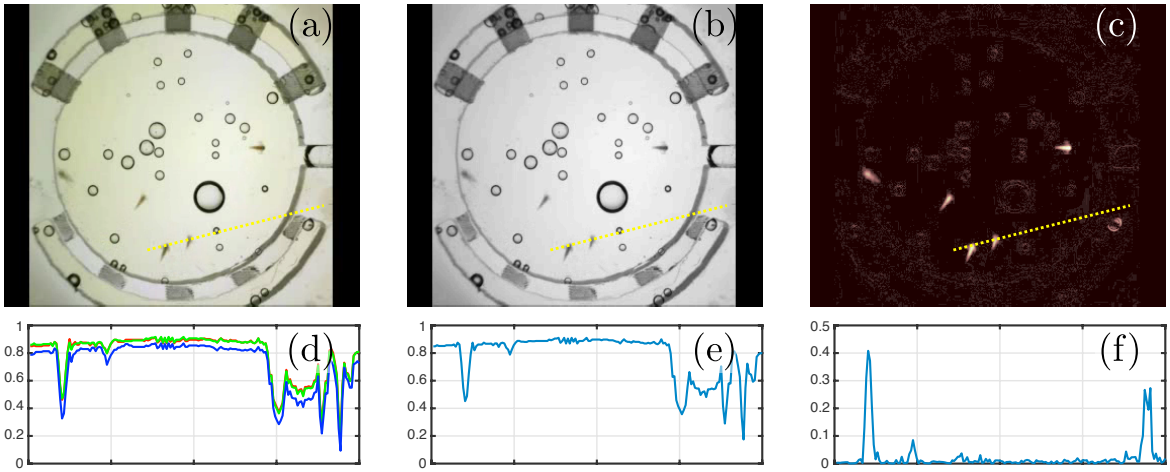


Figure 3.38 Preprocessing stage on the *Artemia Franciscana* dataset. (a) Original RGB image with highlighted profile (yellow, dotted). On the top row, the images shown are the original (a), the conversion to greyscale (b), and the resulting image after the preprocessing stages of subtracting the mean from all the images (c). On the bottom row, the profiles from the same highlighted lines in (a-c) are shown for all images.

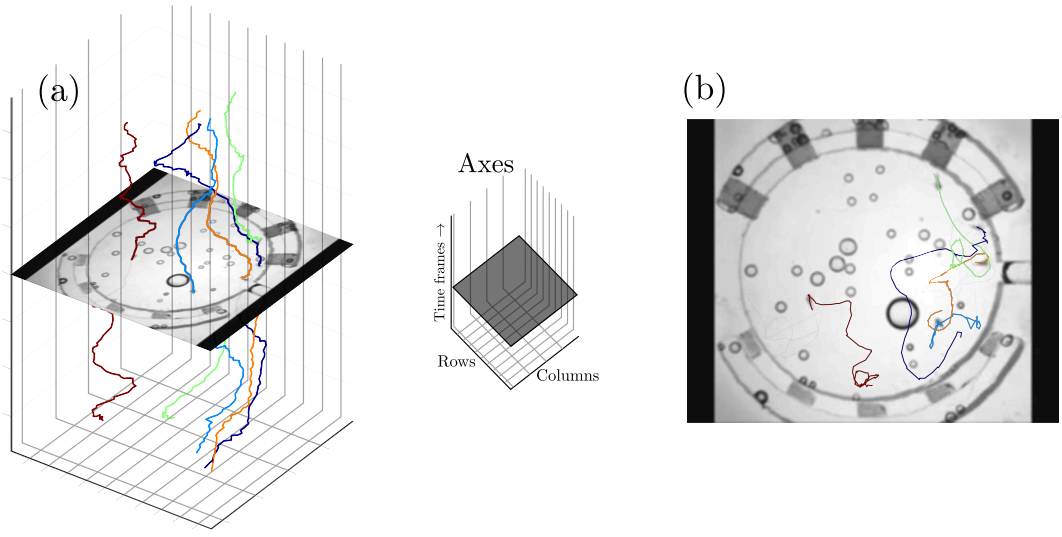


Figure 3.39 Implementation of PhagoSight on *Artemia* data. In the results, the x and y axes represent the rows and columns of the image and the time frames are read from the bottom of each plot to the top. A guide for the axes is presented in the middle of the picture.

Implementation of PhagoSight on Red channel or `MACROS` data. To explore the macrophages data, tests were performed on the `MACROS1` dataset. The results are presented in Figure 3.40, where the red channel was input to PhagoSight. Some problems can be noticed, as some inconsistent tracks can be observed close to the edges of the frames because of problems in the acquisition of images.

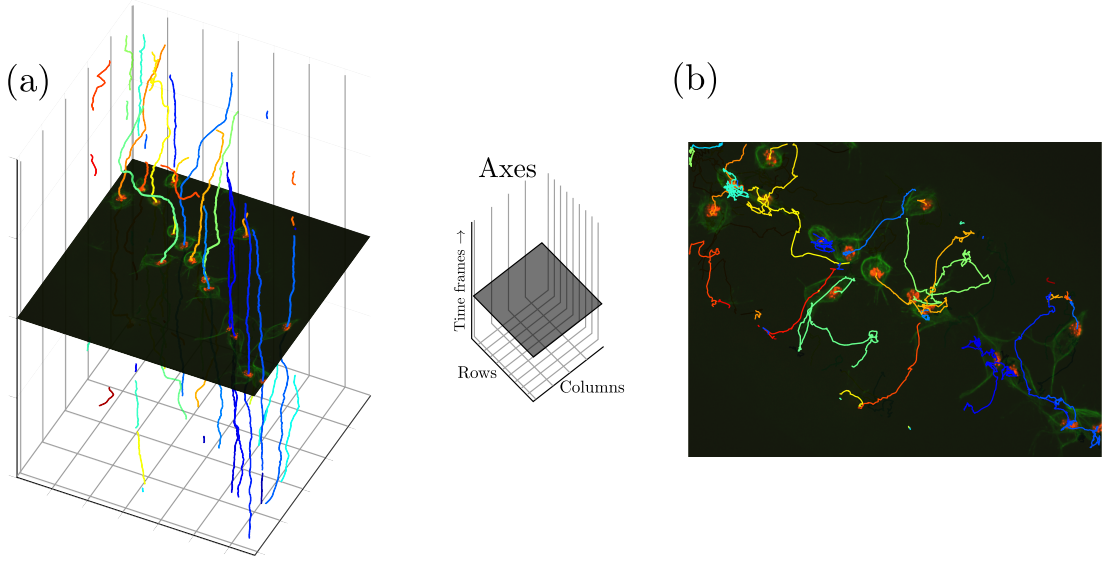


Figure 3.40 Implementation of PhagoSight on Red channel or `MACROS1` data. In the results, the x and y axes represent the rows and columns of the image and the time frames are read from the bottom of each plot to the top. A guide for the axes is presented in the middle of the picture.

3.4 Summary

This chapter described image analysis techniques to segment objects from background. The chapter was utilised as a thorough exploration of the data presented in Chapter 2, in particular the macrophages images. As shown, the macrophages datasets involve several challenges. Most of them are due to the imaging technique used, which caused an uneven distribution of the intensities in the green channel, where microtubules are represented. However, such also allow to observe interactions and cellular structures from a different perspective. Challenges observed included: overlapping due to cell contact and interactions, uneven segmentation and some holes or incomplete segmentation of the cells and inconsistencies in the imaging towards the upper right and lower left corners of the images.

A **traditional image analysis** approach, including preprocessing, thresholding and post-processing, was described thoroughly. Various techniques per stage were discussed. This segmentation approach could overcome some of the challenges in the dataset, such as noise before and after applying a threshold. However, it is not enough to solve the problem of overlapping, since a threshold will categorise intensities into specific classes. Adaptive thresholds did not prove more useful than hysteresis thresholds and had the downside of being time consuming.

Table 3.5 Main conclusions from the data exploration with traditional image segmentation. The methods underlined stood out delivering the best performance for the data observed.

Preprocessing	
Box filter	Both filters produce similar results useful for reducing noise.
Gaussian filter	
Disk filter	Filter would highlight circular structures such as cell nuclei.
Order statistic filters	Nonlinear approach causing distortion and loss of shapes.
Thresholding	
Single	Sensitive to the changes of intensities in the cells.
Hysteresis	Speckle noise is highly reduced.
Bradley and Roth [83]	Sensitivity parameter difficult to optimise per frame.
Pappas [82]	Time consuming. Results marginally better to hysteresis.
Postprocessing	
Opening	Allows for detection of the overall shape of the cells.
Closing	Some unwanted structures get joined to the segmentation, altering the shape detected.

Approaches involving **feature detection** did not produce satisfactory results, which would overcome hysteresis thresholding's outcomes. The case presented for Gabor filters would require more analysis into which wavelengths and orientations would be suitable to successfully classify the different pixels as foreground and background; however this would get the images to the same state where hysteresis thresholding would. Another approach involved the utilisation of SURF features, to try and create a transformation of a cell from a state where it is not overlapping to a state where it is. However, in the experiment attempted, unsatisfactory results were obtained as not enough points could be matched between the cell in frame t and the cell at frame $t+1$.

Finally, three approaches for segmenting cases of overlapping cells were applied. First, a naïve approach using Voronoi Tesselations, would provide a point of comparison for any other algorithm trying segmentation. Then, an Active Shape Model (ASM) and a Level Set Method variation. In the **ASM**, models incorporating all the cells in the training frames would produce a model with eight modes which was hard to manage, since the shapes were extremely varied in shapes. A model incorporating only two overlapping cells was applied, which produced three modes of variation. Unfortunately, in all cases, the selection of landmark points could only be implemented in a way which made the implementation difficult. Possible implementations of this model might work if the landmark points were selected differently and the temporal component was utilised. The idea of exploiting the temporal component is explored in Chapter 5. The **LSM** implementation presented issues with the implementation due to the assumptions of the shapes and sizes of the cells. This implementation, however, provided the idea to improve the detection of boundaries intersections by analysing the border of the overlapped *clump*. Such an idea is explored in the following chapter. While using machine learning techniques was considered, it was not implemented due to the lack of ground truth data, and the difficulty of selecting an objective output for the segmentation. A simple distinction between foreground and background would not require a machine learning approach.

Through the exploration of segmentation techniques, hysteresis thresholding proved to be the most efficient and cost effective. Thus passing the problem into new techniques, which continue from the segmentation of single cells and overlapped clumps. At this point, the analysis of single shape or overlapping boundaries will be introduced. The materials developed in section 2.3 will be exploited to produce a new tool for the analysis of the geometry of the shapes detected, overlapped or not.

Chapter 4

Junction Detection and Segmentation in Single Frames

In this work, the shape of macrophages is considered to be profoundly related to their movement. In this chapter, a tool for the analysis of the junctions is introduced, allowing to describe the shape, by determining whether the object of interest is a clump or a cell. The tool originally was developed to detect specific points on the boundary of an object, called junctions, however it could provide a roadmap for a broader analysis of the shape in a 2D object.

In this chapter, a novel approach to find **relevant junctions** is described in detail from the definition made in the author's own work [71, 72]. From the data presented in Chapter 2, two types of junctions can be recognised in the boundary of an object. Initially, cells elongating as they migrate cause some pointy edges, which can be observed. Situations where macrophages may overlap, result in their boundaries to intersect. Such points can be detected in the boundaries of the clumps that form after segmentation. For segmentation, a **baseline** algorithm is introduced in this chapter, designed to automatically detect the objects of interest in both channels individually. *Clumps* are thus defined as detections in the green channel containing two or more nuclei. A novel tool, ***anglegram*** is presented, which enables two types of relevant junctions to be detected from the boundary of segmented objects. The methodology in junction detection, features and limitations will be presented using tests on synthetic and macrophages data.

Finally, cases of **separating overlapping cells** in single frames using the *anglegram* are presented. First, a naïve approach is presented as a starting point, which only relies on the junctions detected by the *anglegram*. The chapter continues with two

approaches to separate the cells in order to solve the ambiguity introduced by the baseline segmentation. The approaches performed include (i) a boundary analysis of the clump and the underlying edges close to the junctions; and (ii) a custom self-organising maps (SOM) approach which would adapt to the area of the cells lost by the baseline segmentation. All results are compared to overlapping cells separated by the Voronoi approach described in Chapter 3.

4.1 Junctions

Junctions are the intersection of two edges corresponding to points of interest in the shapes of objects and could aid in the analysis of shape. On the boundaries of objects, junctions are commonly acquired by looking for extrema in the curvature of the image gradient [103, 104]. However, a more intuitive approach to defining junctions includes the notion of the tangent. The notion is summarised by the following definition.

Definition 3 (Junction, general) *Let* $c = (x(s), y(s))$ *be the* ***boundary of an object****, where* $s \in [0, 1]$ *and* $c(0) = c(1)$ *. A junction is defined as a point in which the tangent of* c *is undefined.*

Figure 4.1 showcases the two types of junctions and the nature of their generation from the dataset. Preliminarily, both types of junctions can be defined based on the definition above. However, the shapes can be classified based on the state of the objects forming the boundaries. Corners occur due to overlapping of two objects, while peaks occur through the inherent movement and elongation of the cells.

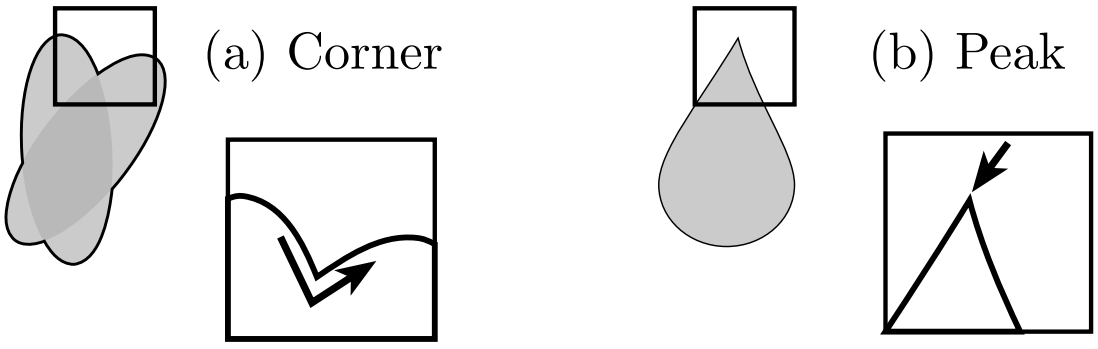


Figure 4.1 Diagram representing the two distinct types of junctions recognised in the working datasets. (a) Corners correspond to junctions that are created by the combined outline of two or more overlapped objects, forming a *clump*. (b) Peaks correspond to elongation of a boundary of the object into a point. Both junction types can be recognised in the macrophages dataset, and were modelled in the synthetic data described in section 2.3.

Two types of junctions can be recognised, however, the previous definition, although intuitive, is too broad. Also, the definition would be difficult to apply in digitalised shapes. In a more accurate description[1]: (i) **corners**, which correspond to junctions found in the boundary of the union of two **overlapping objects**, whose intersections would correspond to the junctions acquired; secondly, (ii) **peak** , which correspond to the pointy edges in **single** objects. The **peaks** detected, and the methodology to do so would provide a tool for the analysis of shape. On the other hand, the **corners** would be used as the basis for completing a segmentation of the overlapping cells.

4.2 Baseline segmentation

This section describes the algorithm to detect the nuclei in the red channel, as well as the cells -or clumps- in the green channel in the `MACROS<ID>` datasets. The algorithm serves as a starting point for additional processing involving (i) the disambiguation of *clumps* in the green channel in single frames, Section 4.6, (ii) the shape tracking, Section 5.2 and (iii) analysis of cell movements, Section 5.3.

As reviewed in Chapter 3, sophisticated algorithms have not produced satisfactory results, furthermore, there is not enough annotated data to produce a machine learning model for the shape of the cells that would allow for the segmentation of overlapping cells. Therefore, the algorithm developed will favour simplicity and efficiency allowing for new methods like the *anglegram* [72] to have a robust foundation. Figure 4.2 shows a diagram of the algorithm used to segment single cells and *clumps* from the background. The algorithm is straightforward and easy to understand. Each channel was treated independently as the intensities distributions of both channels are different; the same methodology was used for both. First of all, a **preprocessing** stage is done through spatial smoothening, with a disk averaging filter of radius $r = 3$ for the red channel and a Gaussian filter, standard deviation $\sigma = 1$ and size of 7×7, for the green channel. The **segmentation** stage is performed by a hysteresis threshold with the two levels chosen by the Otsu algorithm, as shown in equation (B.12). The Otsu thresholds were computed individually per channel. Finally, **postprocessing** was performed through a morphological hole filling, followed by an opening with a structural element $\mathbb{S}$ of a disk of radius $r = 3$.

[1] Note that the notation utilised in [72] is different, but the notation could lead to confusions. A more similar notation can be found in [71].

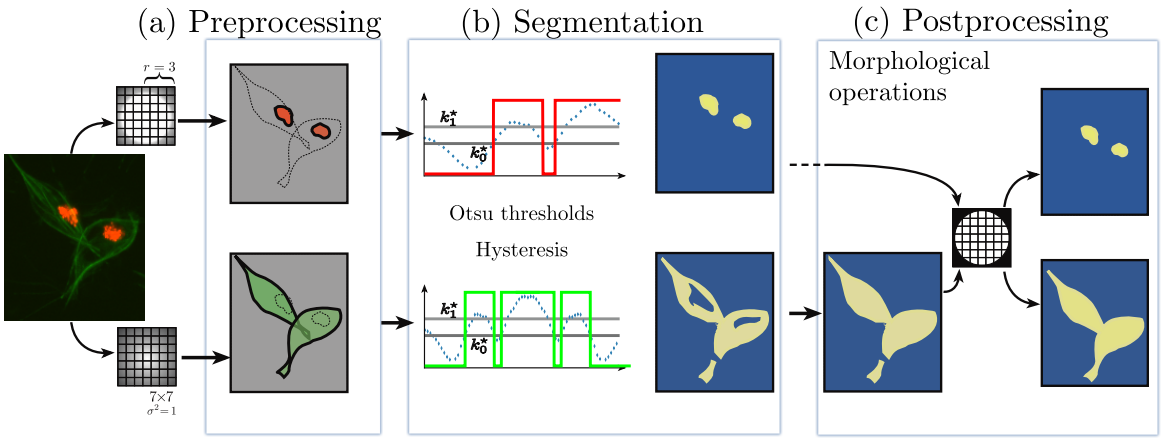


Figure 4.2 Diagram of baseline segmentation. As overviewed in chapter 3, detection of objects is made in three steps. Each channel in the images is treated independently from each other. (a) Preprocessing is done through spatial smoothening, with a disk averaging filter ($r = 3$) for the red channel and a Gaussian filter $\sigma = 1$ for the green channel. (b) Segmentation is done by a Hysteresis threshold with the levels chosen by the Otsu algorithm. (c) Postprocessing for the green channel involved a morphological hole filling, then an opening with a $\mathbb{S}$ of a disk ($r = 3$).

Experiments with baseline segmentation

The baseline segmentation technique was evaluated and compared against ten consecutive frames with ground truth. The ground truth was obtained through the `freehandSegmentation` routine described in section 2.4, where the objects in the ground truth were joined with the union operator. This procedure produced ground truth clumps. The Jaccard index per detected object was calculated for all objects in the consecutive frames with available ground truth, concerning frames 45 to 54 of the `MACROS1` dataset. The results are shown in figure 4.3, where the frame numbers are displayed along the axis. Two arrows point to relevant cases, representing the highest (red) and lowest (black, dotted) Jaccard indexes scores.

Regarding the detection of objects, a qualitative comparison is presented in Figure 4.4, showing the segmentation output compared against ground truth at three distant time frames. The colour code in the comparison column, shown in 4.4(d), is the same as in previous figures throughout this work: **black** represents the background or true negatives, **yellow** represents false negatives, **blue** represents false positives and **red** true positives. It is important to notice that **baseline segmentation** refers to the detection of frames, regardless of overlapping.

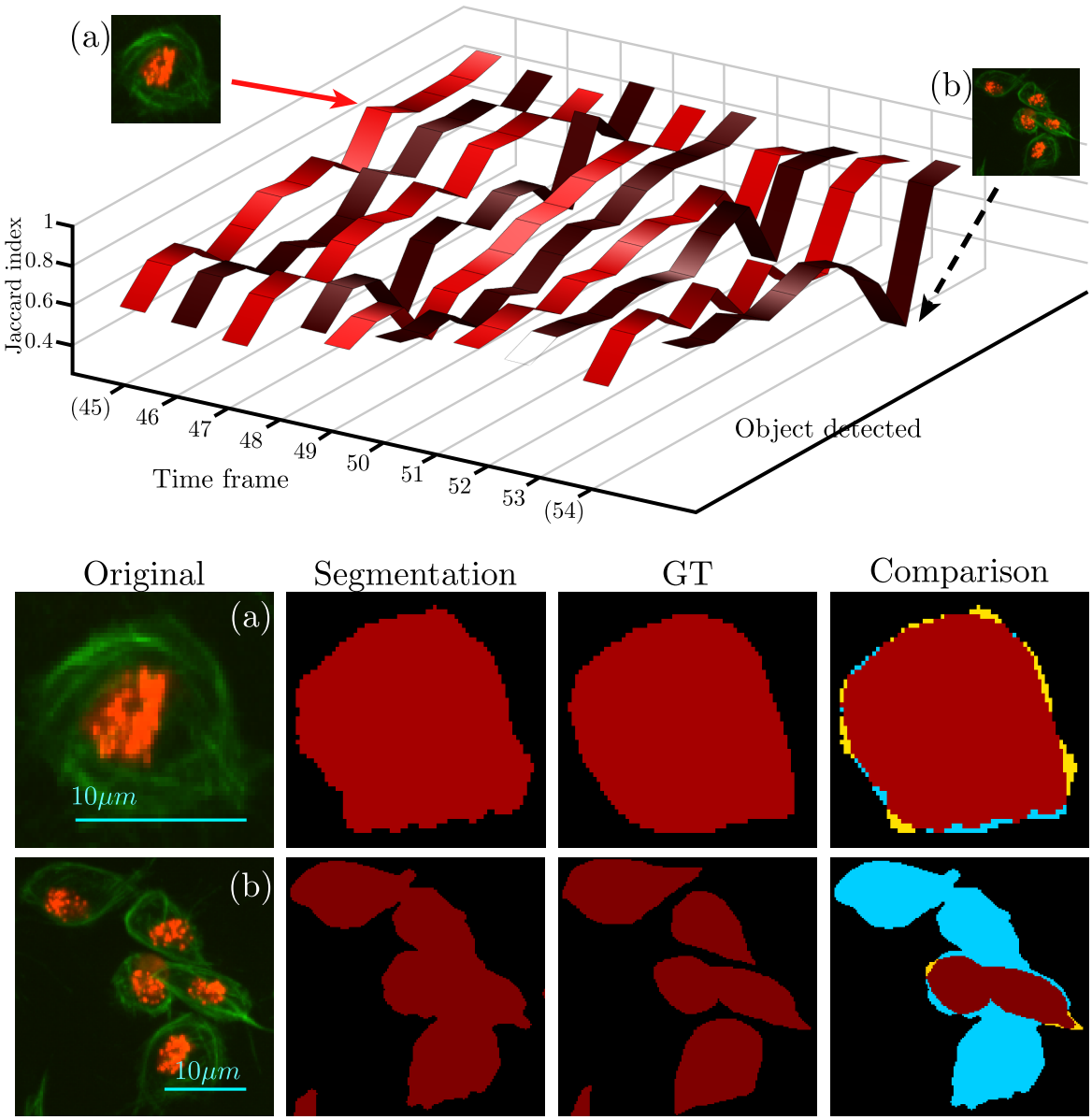


Figure 4.3 Comparison of the Jaccard index for each object detected, whether a clump or not, at each of the ten consecutive frames with available GT. The ribbon plot at the top displays the time frames and the object detected per frame, on the vertical axis, the Jaccard index is shown. Two relevant cases are highlighted: (a) depicts a cell in frame 45 achieving a high Jaccard index (red arrow). (b) depicts a clump detected in frame 54 and its Jaccard index is marked with a black arrow (dotted). In the comparison for both cases **black** is background, **yellow** represents false negatives, **blue** represents false positives and **red** true positives. Notice that the ground truth presents clumps as a result of combining all the manually selected cell outlines.

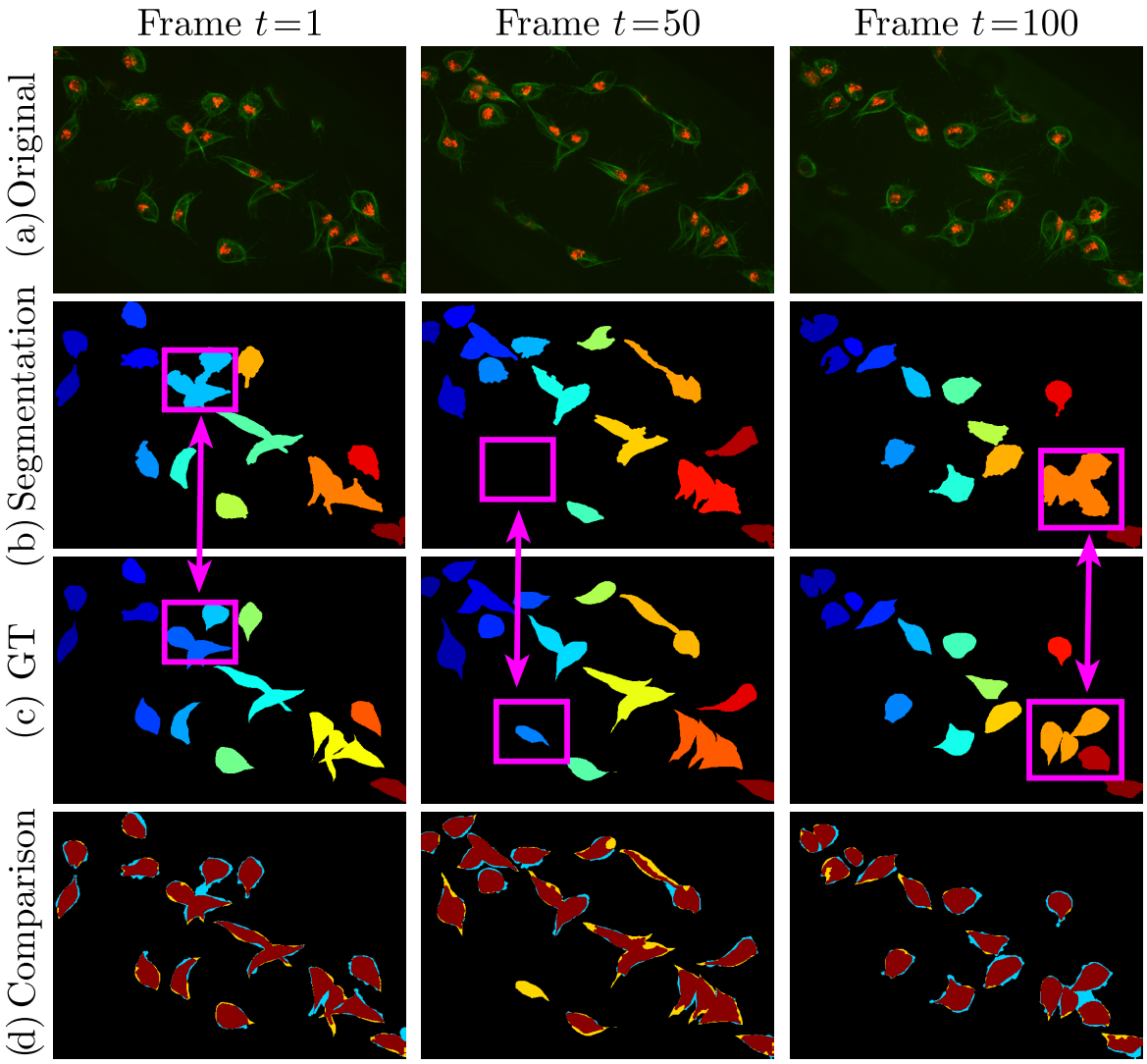


Figure 4.4 Detection of objects in the green channel on three representative frames of the `MACROS1` dataset and a qualitative comparison with the ground truth. (a) shows the original frames in the dataset, while (b) and (c) depict the segmentation output and ground truth, respectively. Subtle differences can be observed, like the number of objects, relating to the number of clumps detected by the segmentation algorithm and the number of actual clumps in the frame. Magenta boxes and arrows are used to highlight relevant differences. (d) shows a qualitative comparison with the ground truth, where **black** represents the background, **yellow** represents false negatives, **blue** the false positives and **red** true positives.

Partition of clumps based in the detection of junctions

Identifying the object boundaries contained in a detected clump could be achieved by detecting the intersections of such cell boundaries. The method will be fully defined in Section 4.3. Given a set of junctions of a boundary known to contain overlapping objects, a number of partitions of the *clump* can be proposed. Such partitions would then be suitable candidates for approximating the original overlapping objects. Selection of such **candidates** would then be a relevant task. This area of the research has been incorporated as motivation to the algorithm for **junction detection**. An example is provided to describe the problem, then in section 4.6, an initial approach is provided in an application.

Considering two overlapping ellipses, the individual boundaries can have either two, three or four intersections. The case for two intersections is solved almost trivially, as there is only one line cutting through the objects and providing the correct candidates. The case for 3 intersections, in practical terms, is not applicable and would not be explored considering digitalised boundaries. Finally, the case for four intersections is of importance, as without more information, eight possible partitions are possible and it is only through some prior knowledge that the candidates can be produced.

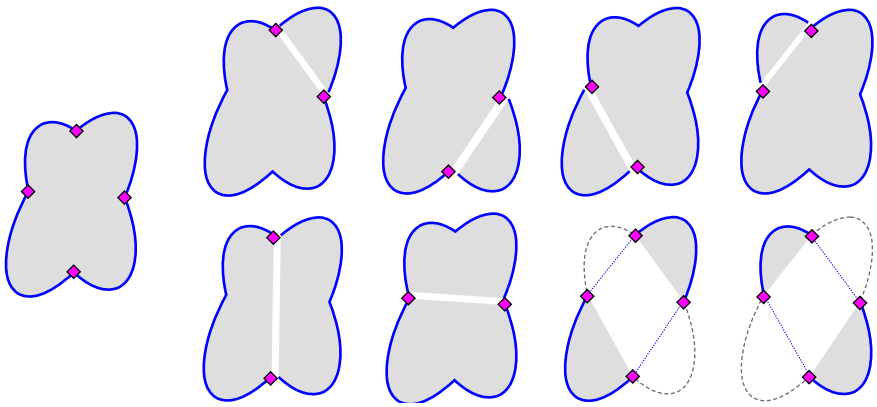


Figure 4.5 Representation of possible partition of a clump based on the position of its junctions. Eight different partitions of the object are possible. The prior knowledge of this experiment is that there are two ellipses overlapped, which makes easy to visually identify which candidates are appropriate.

Consider the case of several corners being detected in the outline of a *clump* boundary. As with the ellipses, only two or four can be correct detections, although due to cell shapes, more could be detected. Thus, a procedure must be considered to identify possible candidates in a clump.

4.3 The *Anglegram* Matrix

The *anglegram* is a matrix associated to the boundary of an object, which allows for 2D shape characterisation; one of its key applications is the detection of relevant junctions. The elements in the *anglegram* matrix are calculations of **inner angles of the points** in the boundary, allowing for a computational description of relevant junctions. The key terminology involves a calculation of various angles per point in the boundary of an object, called *angle variations*. The angle variations allow for a distinction between both types of junctions, and can be used in practical scenarios when dealing with discrete boundary points. Following the definitions, tests of the methodology with synthetic data, as well with images of binarised macrophages will be shown.

Discrete boundary of an object. As mentioned before, object shapes can be modelled with a parametric, closed curve, $c(s) = (x(s), y(s))$, such that $s \in [0, 1]$ and the point $c(0)$ is at the same position as the point $c(1)$. Let $\mathcal{B} \subset \mathbb{R}^2$ define the boundary an object containing N **ordered** points. Then $\mathcal{B}$ can be understood as the set described in equation (4.1):

$$\mathcal{B} = \left\{ \mathbf{x}_i = (x(s_i), y(s_i)) \,\middle|\, s_i = \frac{i}{N}, \quad i = 1, \cdots, N \right\}, \tag{4.1}$$

where $\mathbf{x}$ corresponds to a specific vector or point with two coordinates $\mathbf{x} = (x, y)^T$. Thus, each of the points $\mathbf{x}_i \in \mathcal{B}, \quad i = 1, \cdots, N$ in the boundary are ordered. Without loss of generality, the points will be deemed to be ordered in a clockwise manner along the boundary of the object. For example, for a boundary having N points, $\mathbf{x}_{N+i} = \mathbf{x}_i$. From Section 2.3, a *clump* of two objects will correspond to two boundaries partially overlapped.

Angle variations on boundary points

Angle variations refer to multiple measurements of angles which can be computed from a single point in the boundary of an object. In mathematical terms, in a 2D space, an angle can be measured through three points: a middle point which serves as a reference for the other two points. Classical mathematical tools, can then be used to compute the value of the angle. To describe the angle variations of a shape, the inner angle of a point will be defined by using each point along the boundary as the reference point, and selecting two angles in the boundary to perform the calculation.

Definition 4 (Inner angle of a point) *The inner angle of a point* $\mathbf{x}_i \in \mathcal{B}$ *in the boundary is the angle* $\theta_{i,j}$ *adjacent to the point, and measured from the jth previous point* $\mathbf{x}_{i-j}$ *to the following jth position* $\mathbf{x}_{i+j}$*. The angle is then depicted as* $\theta_{i,j} = \measuredangle \mathbf{x}_{i+j}\mathbf{x}_i\mathbf{x}_{i-j}$.

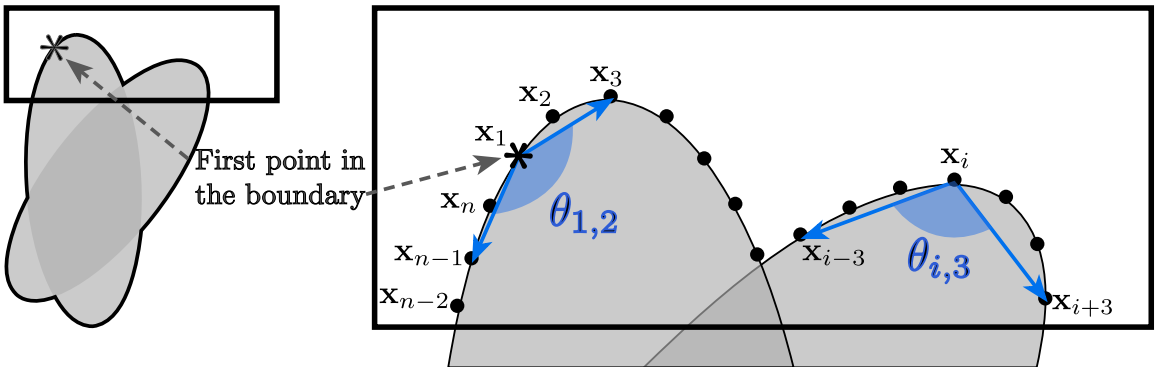


Figure 4.6 Illustration of the arrangement of n points in a boundary and the calculation of the inner angles of two points. Notice the circular arrangement of the points meaning that $x_{n+i} = x_i$.

Figure 4.6 shows the circular arrangement of points in a given boundary, as well as the selection of the three points necessary to compute the inner angle of each point. As the points in the boundary $\mathcal{B}$ correspond to a discretisation of a continuous curve $c(s)$, the N points in the boundary can be thought of as circular, or cyclical. This is important when measuring the inner angle of certain points at certain separation distances.

For simplicity, the points at a separation j are depicted as $\mathbf{x}_{i-j}, \mathbf{x}_{i+j}$, however it can be seen from the figure that the at a point $\mathbf{x}_i$, the selection of points $i \pm j$ will correspond to the modulo, $\mod(i \pm j, N)$. From the last statement, it can be seen that if $i + j > N$ then the point $\mathbf{x}_{i+j} = \mathbf{x}_{i+j-N}$; conversely, if $i - j < 1$, then the point $\mathbf{x}_{i-j} = \mathbf{x}_{i-j+N}$.

The mathematical derivation of the calculation of the *inner angle of point* $\mathbf{x}_i \in \mathcal{B}$, with separation j, is presented. The angle is computed by the translation of the two adjacent points $\mathbf{x}_{i\pm j}$ to have the $\mathbf{x}_i$ in the position of the origin. Then, the calculation of the angle is made by the calculation of the angle between the resulting vectors. Let $\mathbf{p}_i \in \mathcal{B}$, the jth previous and following positions in the boundary are $\mathbf{p}_{i-j}, \mathbf{p}_{i+j} \in \mathcal{B}$, respectively. Let the vectors $\mathbf{v}_+, \mathbf{v}_- \in \mathbb{R}^2$ be defined as $\mathbf{v}_+ = \mathbf{p}_{j+i} - \mathbf{p}_i$, and $\mathbf{v}_- = \mathbf{p}_{j-i} - \mathbf{p}_i$. The angle, θ_v, adjacent to the origin and measured from $\mathbf{v}_+$ to $\mathbf{v}_-$ will be equal to the inner angle of the ith point, $\theta_{i,j}$. Figure 4.7 contains a graphical explanation of the detailed implementation of such a calculation. Finding the angle θ_v is done through Equation (4.2). Given the previous definition, a more accurate

description of the types of junctions is outlined in Definition 5.

$$\cos\theta_v = \frac{\mathbf{v}_+ \cdot \mathbf{v}_-}{\|\mathbf{v}_+\|_2 \|\mathbf{v}_-\|_2}. \tag{4.2}$$

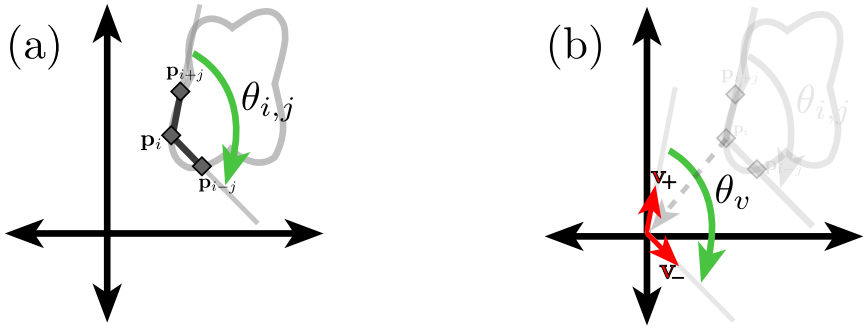


Figure 4.7 Graphical representation of the calculation of the inner point angle of point $\mathbf{p}_i$ at separation j. **Left.** representation of the inner angle of point $\mathbf{p}_i$ at separation j. Notice that the points in the boundary are taken in clockwise order. **Right.** representation of the translation vectors $\mathbf{v}_+, \mathbf{v}_-$. Full explanation in text.

Definition 5 (Corner and peak junctions) *A **corner** is a junction in which the inner point angle for most separation distances is greater than 180 degrees. Conversely, a **peak** is a junction in which most of its inner point angles are* acute, i.e. *less than 180 degrees.*

Notice how the definition above does not contain assumptions about the closed boundary. A junction is, thus, a point where the inner angle will range between certain values, at various separations j. The discrete curves representing the boundaries would complicate the detection of junctions, due to quantisation noise. Figure 4.9 shows examples of the calculation of an *inner angle* for a given point in a clump boundary, the calculation is the same for single objects. Next, the definition of the anglegram matrix will provide a way to select corners or peaks based on calculations made upon it. For simplicity, the detection of junctions will be described for the case of corners, following [72]. Detecting peaks is a similar procedure, thus modifications to the method required will be outlined. By visual inspection, the inner angle of a junction would be greater than 180 degrees for a number of separations j. This number of separations will be referred to as the *depth* of the junction. Thus, the method consists of computing the inner angle $\theta_{i,j}$ at every point $\mathbf{p}_i \in \mathcal{B}$, and on every separation j.

The Anglegram matrix

The anglegram [72] is a tool to determine the relevant junctions of a particular shape, based on its boundary points. The *Anglegram matrix* is defined as follows.

Definition 6 (Anglegram matrix) *Let the boundary $\mathcal{B}$, be referred to as the set containing its points $\mathcal{B} = \{\mathbf{x}_i\}_{i=1}^{N}$, with the N ordered points corresponding to an object. For each point $\mathbf{p}_i \in \mathcal{B}$, the neighbours at the jth separation distance are the points $\mathbf{p}_{i-j}, \mathbf{p}_{i+j} \in \mathcal{B}$. Then, the* ***anglegram matrix*** *$\Theta(i,j) = \theta_{i,j}$ is defined as the values of the inner angles of each point i and per separation j, that is $\theta_{i,j} = \measuredangle \mathbf{p}_{i+j}\mathbf{p}_i\mathbf{p}_{i-j}$, calculated with equation* (4.2).

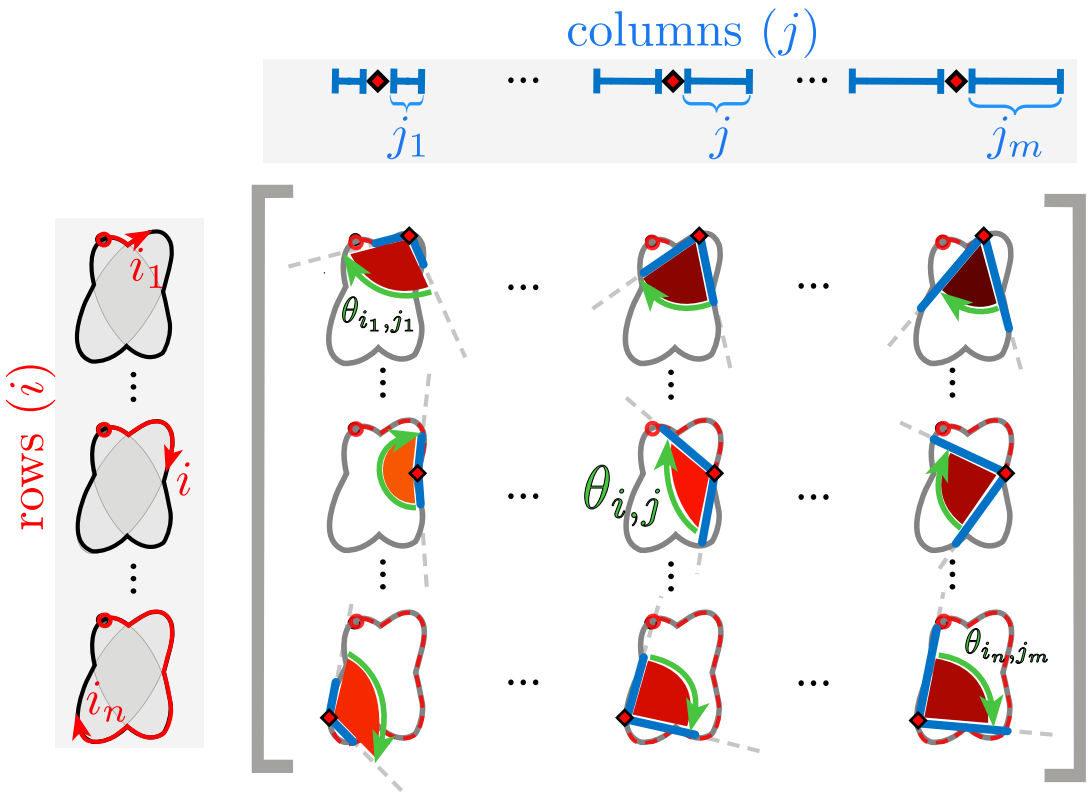


Figure 4.8 Explanation of inner angle of a point in the construction of the anglegram. The diagram shows a representation of nine arbitrary entries $\theta_{i,j}$ of the *anglegram matrix*. Each entry corresponds to an *inner point angle* at a specific *separation*. In the diagram, as in the matrix, the rows (i) correspond to a single point alongside the boundary (red ⋄) that start at a specific point (marked ○); the columns (j) correspond to the separation from the point i⋄ and from there the angle is taken. Each corresponding entry, $\theta_{i,j}$, of the *anglegram* matrix Θ is marked with a green arrow, furthermore, each angle is shaded to match the colour map used in the anglegram in figure 4.9.

The definition considers the concepts of **inner angle of a point**, i and **separation distance**, j for the construction of the anglegram; the columns represent the separation distance along the boundary and the rows represent each point in the boundary. The *anglegram* matrix would provide a general overview of the shape and the inner angle of the points along the boundary. Figure 4.8 represents the calculation of each inner point angle based in some examples, each angle is shown in a different colour to symbolise the size of the angle. Furthermore, Figure 4.9 contains an example of this construction in a synthetic case, where (b) shows a single row of the anglegram, whose values correspond to a single point (blue ∘) and three instances of the inner point calculation at different separations (red+, yellow ∇ and green ⋄).

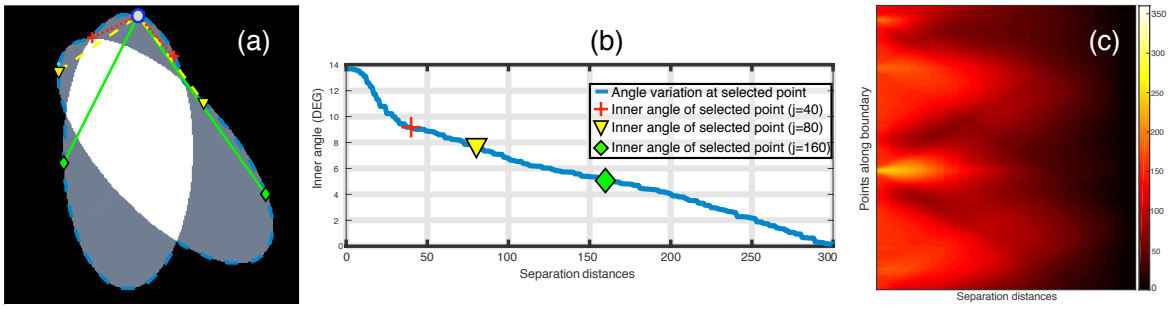


Figure 4.9 Representation of *inner point angle* calculation and generation of *anglegram matrix*. (a) Represents a synthetic clump with its boundary outlined (blue - -), where a point (blue ∘) in the boundary will have various inner point angles per separation j. All the inner point angles for the highlighted point are displayed in (b). (c) Shows the *anglegram matrix*, where each row represents the graph displayed in (b) for each boundary point.

Notice that the sampling of the boundary would impact the generation of the anglegram, and the subsequent detection of joints. As the boundaries are generated using the MATLAB® function `bwbondaries`, they will return a number of points corresponding to the number of boundary pixels for the corresponding region. In developing the anglegram matrix, common number of boundary points sampled, $N = |\mathcal{B}|$, would be ranging from 250 to 300 points, thus producing a matrix with such number of rows and $N/2$ columns. The following sections describe the junction detection algorithms, which use the anglegram. The number of sampling points was only reduced when considering peak-detection.

4.4 Junction detection with the *Anglegram*

In this section, the process to find relevant junctions on the boundary of an object by using the *anglegram* is presented. The method will employ the definitions of **corner** and **peak** and utilise them to search for extrema in the *anglegram* matrix. As any junction will correspond to a point in the boundary, the inner point angles of it can be assessed by looking at a projection of the rows of the *anglegram*. The method is similar in both types of junctions, as shown in figure 4.10. After generating the *anglegram*, an intensity projection can be applied to its rows where each data point would correspond to the positions in the boundary.

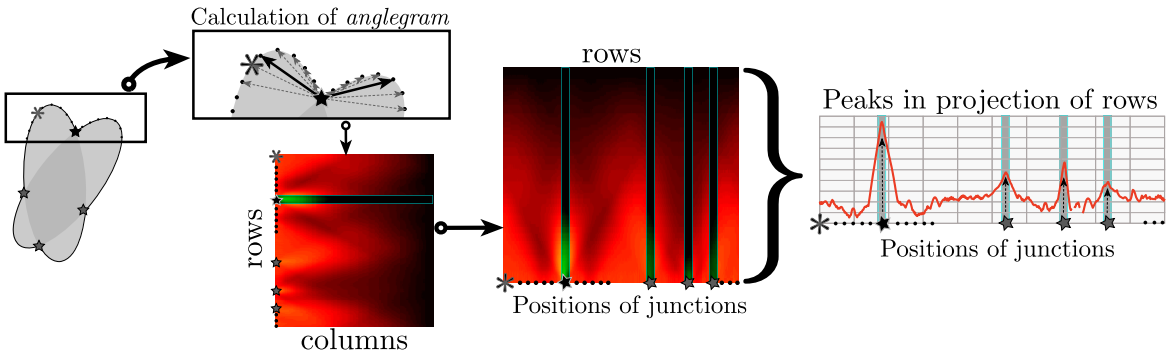


Figure 4.10 Graphical representation of the detection of junctions via the *anglegram*. In this example which represents a boundary with corner junctions, large inner point angles are searched for. Therefore, the maximum intensity projection of the rows is represented. Each junction is represented with a star ($\star$) marker, also shown in the corresponding row in the *anglegram* and position in the projection, where peaks display the detection of a junction.

Determination of the projection over the rows of the *anglegram* As the local extrema on a projection over the horizontal dimension of the anglegram is related to the position of the junctions of the boundary and the depth of the junction. Each row, $\Theta(i,:)$, corresponds to the inner angles of point $\mathbf{p}_i$, therefore taking a summary of the rows would yield a measurement of the general inner angles of each point. For the detection of corners, the maximum intensity projection, MIP, was compared with the mean, median and area under the curve. Figure 4.11 displays the four types of projections used for a particular boundary. The maximum intensity projection produces a particular shape in which the higher levels appear at positions in the boundary where the junctions are located.

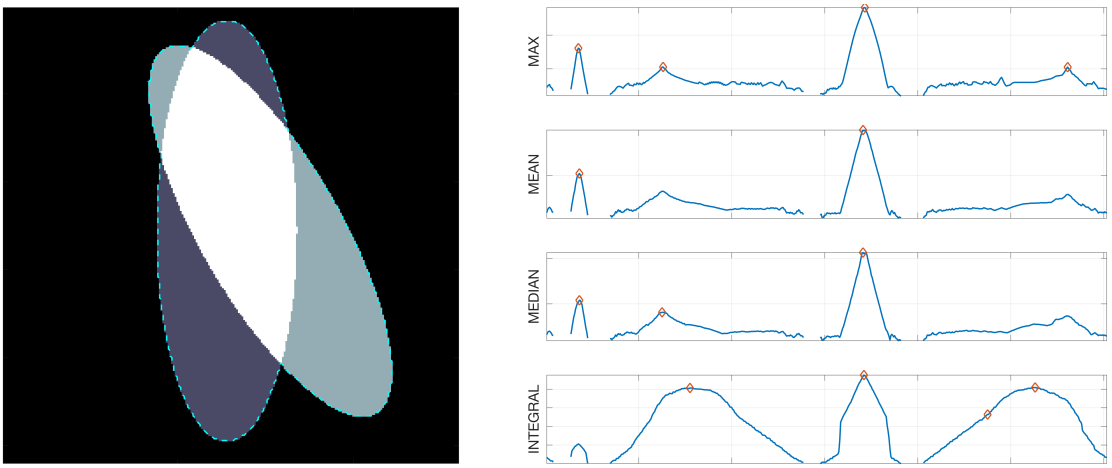


Figure 4.11 Tests from the different row projections of the *anglegram*. On the left, the overlapping object presents four corners. On the right, the different projections. Diamond markers of relevant peaks are displayed in the different curves. The maximum intensity projection consistently outperformed the other measurements when detecting corners, whereas the mean and median would often miss junctions with small depth and the area under the curve would produce false detections.

Detection of corners

The *anglegram* matrix, Θ, contains the inner point angles calculated at every point, and at all separation distances. Thus, every row in the anglegram corresponds to the measurements made at a specific point in the boundary. In the case of detecting corners, definition 5 specifies that corners are junctions whose inner angles are consistently obtuse. As seen in Figure 4.10, a projection over the rows of the *anglegram* Θ is needed to find the positions along the boundary that contain a junction. And as seen in Figure 4.11, the maximum intensity projection (MIP) over the rows of the *anglegram* was chosen; this vector is represented by the symbol $\hat{\theta}_{Pr}$.

The *anglegram* needs to be preprocessed to account for quantisation errors in the boundaries extracted from the clumps, especially at close separation distances. First of all, an averaging filter of size 5×5 was applied to the anglegram matrix, Θ, before the calculation of $\hat{\theta}_{Pr}$. The second step ignores the first columns of the *anglegram*. The local maxima of the 1D projection were found by using the function `findpeaks` from MATLAB®, which identifies local maxima of the input vector by choosing points of which its two neighbours have a lower value. Due to quantisation noise in $\hat{\theta}_{Pr}$, the parameters `MinPeakDistance` and `MinPeakHeight` were set to empirically consistent values. First, `MinPeakDistance`, which restricts the function to find local maxima with

a minimum separation, was set to 25. Furthermore, the parameter `MinPeakHeight` was set to $\text{mean}(\hat{\theta}_{Pr}) + 0.75 \times \text{std}(\hat{\theta}_{Pr})$.

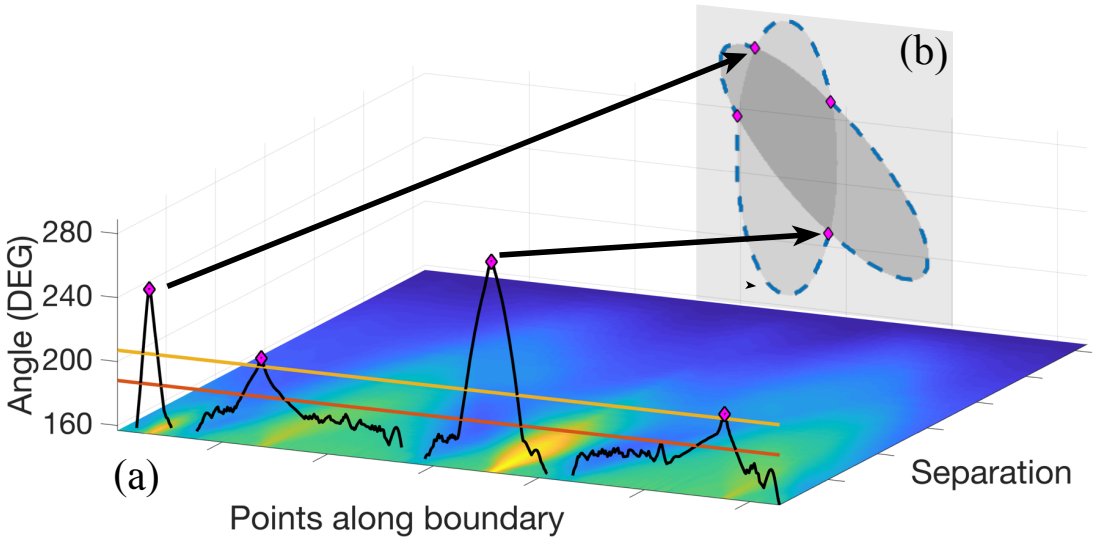


Figure 4.12 Junction detection on overlapping objects through the maximum intensity projection of the anglegram matrix. The junctions detected on a synthetic pair or ellipses is shown in (a), where the boundary of the clump is represented as a dotted line (blue) as well as the junctions (magenta $\diamond$). The definition of $\hat{\theta}_{Pr}$ is represented in (b), where the anglegram matrix Θ is displayed in a plane and $\hat{\theta}_{Pr}$ is represented along the boundary points. Detection of junctions are shown with $\diamond$ markers (magenta). Notice the two horizontal lines representing mean $\hat{\theta}_{Pr}$ and mean $\hat{\theta}_{Pr} + 0.75\,\text{std}\,\hat{\theta}_{Pr}$.

Detection of peaks

The detection of peaks, is analogue to the detection of corners. The process is similar to the process in Figure 4.10. In this section, the alterations made to the process are described. To detect peaks, two main alterations were implemented: (i) resizing the anglegram to have 64 rows to reduce noise, and (ii) taking the *minimum* intensity projection (mIP) of the first half columns, as the final columns are lower by the definition of the inner point angle measurements. An extra condition was added to avoid over-detecting peaks in circles: when detecting over four peaks with a mean value greater than 120 degrees, the peaks detected are discarded.

The process of detecting peaks involves getting the minimum intensity projection (mIP) of the *anglegram* and finding the points which correspond to the local minima of the mIP, also referred to as $\hat{\theta}_{Pr}$. The procedure to find the local minima is analogue to the process of detecting corners, as the same MATLAB® function findpeaks($-\hat{\theta}_{Pr}$)

is used. Figure 4.14 displays the minimum intensity projection of the anglegrams computed from the mean basic shapes in figure 4.13.

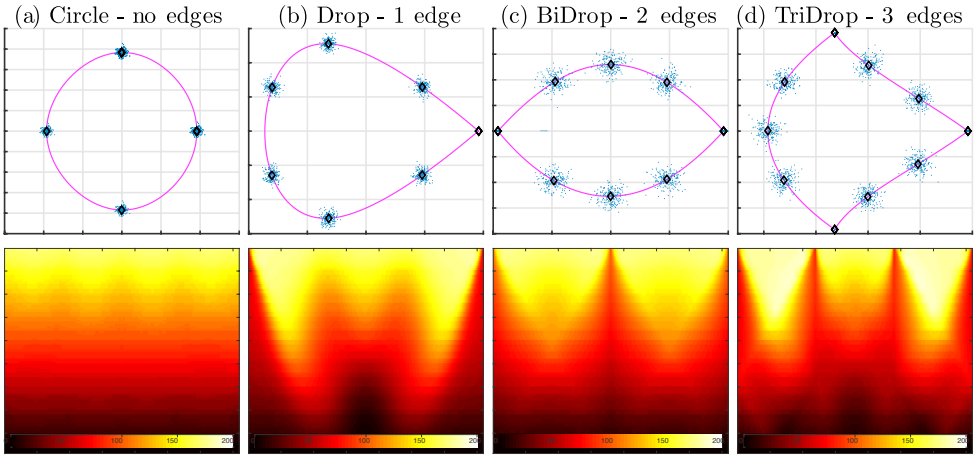


Figure 4.13 Anglegram of basic shapes. As presented in Figure 2.21, two hundred instances of each basic shape were generated. **Top row.** The four mean basic shapes. In each shape, the generated points is presented (·, blue) along its corresponding centroids (◇, black). **Bottom row.** The *anglegram* was computed for each mean shape. The matrix has been transposed to show the boundary in the horizontal axis.

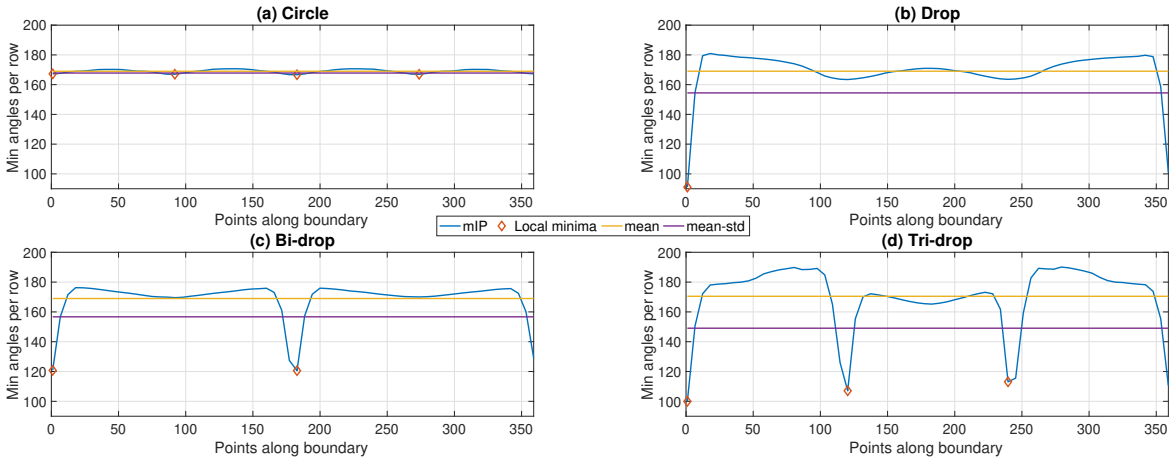


Figure 4.14 Minimum intensity projection of the anglegram of basic shapes. (a-d) show the resulting curves when computing of $\hat{\theta}_{Pr}$. The strong blue lines represent the mIP, $\hat{\theta}_{Pr}$, the red marker (◇) represent the local minima found through the function findpeaks. The yellow and purple horizontal lines correspond to the values of $\text{mean}(\hat{\theta}_{Pr})$, and $\text{mean}(\hat{\theta}_{Pr}) - \text{std}(\hat{\theta}_{Pr})$, respectively. Notice that in (b-d), there appears to be an extra local minimum at the end of the span of the graph, however, that corresponds to the same peak as the one found in the beginning.

4.5 Experiments with the *Anglegram*

The methods outlined in Section 4.4 were validated through tests in the synthetic data described in Section 2.3 to find its limitations. Then, the method was applied to different instances of the macrophages data both for cases of overlapping, as well as different instances of single cells.

Tests in corner detection

Two experiments were designed to test the functionality of corners. The first involves the detection of corners in different cases of overlapping ellipses with varying angles differences and separations. Then, some experiments of corner detections using instances of the `MACROS1` dataset. It can be anticipated that the methodology as described before will have its limitations with certain ranges of angles. It is important to recall that the maximum intensity projection (MIP) is computed per columns, and the angles below a given threshold are discarded. Thus, cases with angles close to 180 degrees are expected to be missed from the analysis.

Synthetic data Images of overlapping ellipses with varying ranges of angles and separations were analysed. The ellipses generated followed the procedure explained in Section 2.3, where each image would contain two ellipses, a reference one $\mathcal{E}_0$, and an ellipse with varied angle and distance from the centre $\mathcal{E}_{\Delta,\phi}$. In the experiment, the images included a variation of ten different angles ϕ ranging from 0 to 90 degrees with a variation of 17 different separations Δ ranging from 0 to 160 pixels from the centre of $\mathcal{E}_0$. In total, 170 images were generated, of which 32 were removed from the analysis, as these were cases where the separation would produce non overlapping ellipses.

A subset of the images is presented in Figure 4.15. In it, the detected junctions are shown in a green marker ($\circ$) whereas the ground truth for the junctions is shown in a red marker ($+$). Places where a junction was missed, only the ground truth marker would be visible. Based on the corresponding value in the maximum intensity projection (MIP) of the anglegram matrix, given the methodology explained in Section 4.4, the junctions that were correctly detected on the synthetic data had a range of angles [188.64 - 328.4] degrees; whilst the missed junctions had a range of [162 - 191.96] degrees. A correct detection would place the junction within 5 pixels of the known intersection of the boundaries. This indicates that very wide angles, close to a straight

line are easy to miss. Detail of the cases where error was found are presented in figure 4.16, where the previous statement can be assessed.

Table 4.1 Synthetic test of corner detection from 139 cases of overlapping. Rows show the number of corners detected by the algorithm compared to the real intersections.

Intersections / Corners	Corners detected (%)			
	≤ 1	2	3	4
2	14.29	**85.72**	0	0
4	6.25	8.33	4.17	**81.25**

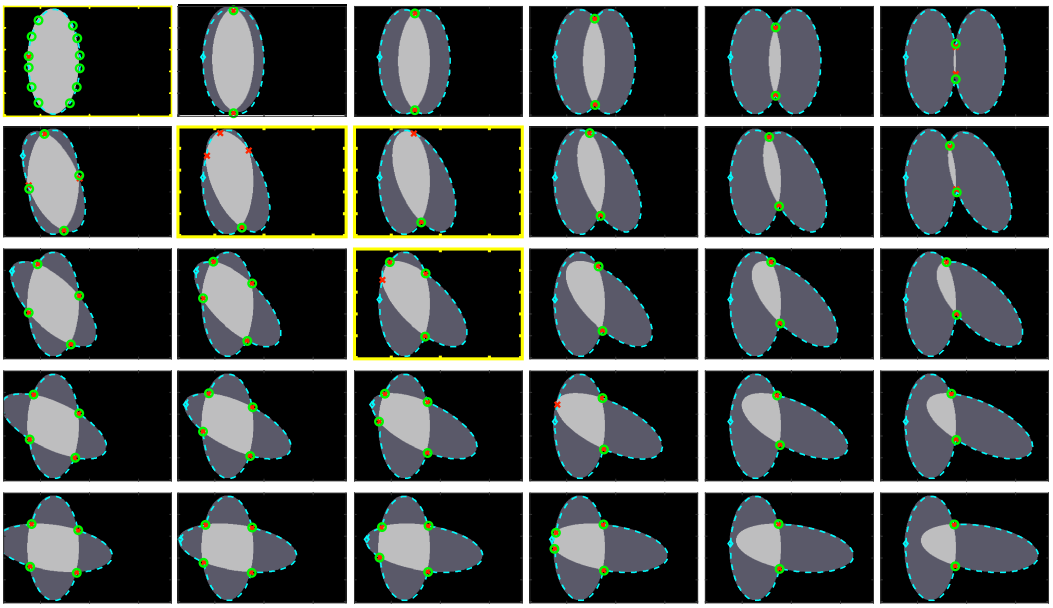


Figure 4.15 Results of the corner detection through the *anglegram* method for overlapping ellipses. The variation of angles and distances can be appreciated. The joint boundary, corresponding to the overlapping region is shown in cyan (- -). In all cases, the detected junctions are presented in a green marker ($\circ$) and the ground truth junctions are shown in a red marker ($+$). Overlapping instances of interest have been highlighted in yellow.

Detection of junctions in macrophages data

The method to find corners was applied to the macrophages dataset. The method requires the boundary of overlapping objects, this requires a segmentation technique to binarise the images and thus produce the boundary $\mathcal{B}$. To ensure that the methodology of detecting junctions would be isolated from the segmentation, the ground truth was

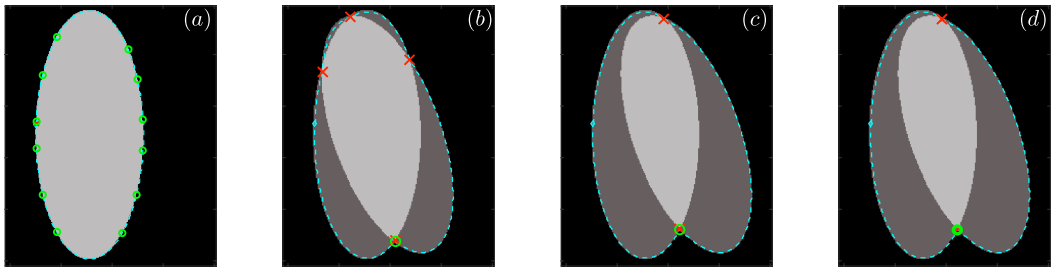


Figure 4.16 Cases where the detection of corners was not accurate. The joint boundary, corresponding to the overlapping region is shown in cyan (- -). In all cases, the detected junctions are presented in a green marker (∘) and the ground truth junctions are shown in a red marker (+). Case (a) is an extreme case, as both ellipses are occupying the same space and have the same size. The junctions in (b-d) were most likely missed due to the inner point angles not being large enough, or as can be seen in (b) caused by a lack of depth of the corner.

used, by merging the two overlapping objects into a single boundary. Eleven frames were segmented manually. The frames showcase four clumps as they evolve in time.

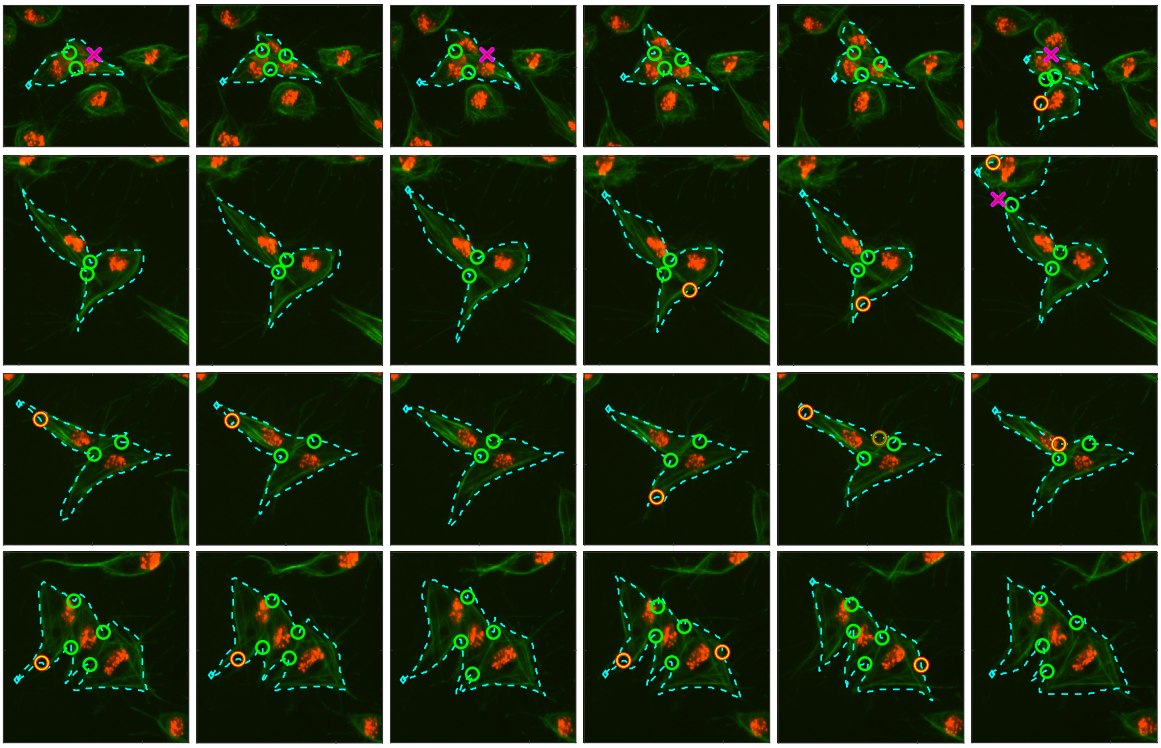


Figure 4.17 Results of the corner detection through the *anglegram* method for clumps in macrophages data. The joint boundary, corresponding to the overlapping region is shown in cyan (- -). In all cases, the detected junctions are presented in a green marker (∘) and the ground truth junctions are shown in a red marker (+). Cases where a detection was made that did not have a junction are green circle markers with a red outline.

Tests in peak detection

To detect peaks, a synthetic test was designed utilising the shapes described in Section 2.3. As observed from the data, four main shapes were identified based in the number of pointy edges they showed: circle, drop, bidrop (or *croissant*) and tridrop. As with the detection of corners, two experiments were designed to test the functionality of peaks. The first correspond to detection of peaks in a large number of artificially generated shapes. Then, some experiments of peak detections were made using instances of the `MACROS` datasets.

A collection of 1000 images of basic shapes was generated with control points joined by cubic splines whose independent variable ranged in $\tau \in [0, 2\pi]$. The tests done on the synthetic data were in the range of 86-95% correct detection, as seen in Table 4.2. The method achieved the highest accuracy in the **bidrop** shape, and the lowest in the **drop** shape. A random sample of the test shapes is shown in Figure 4.18.

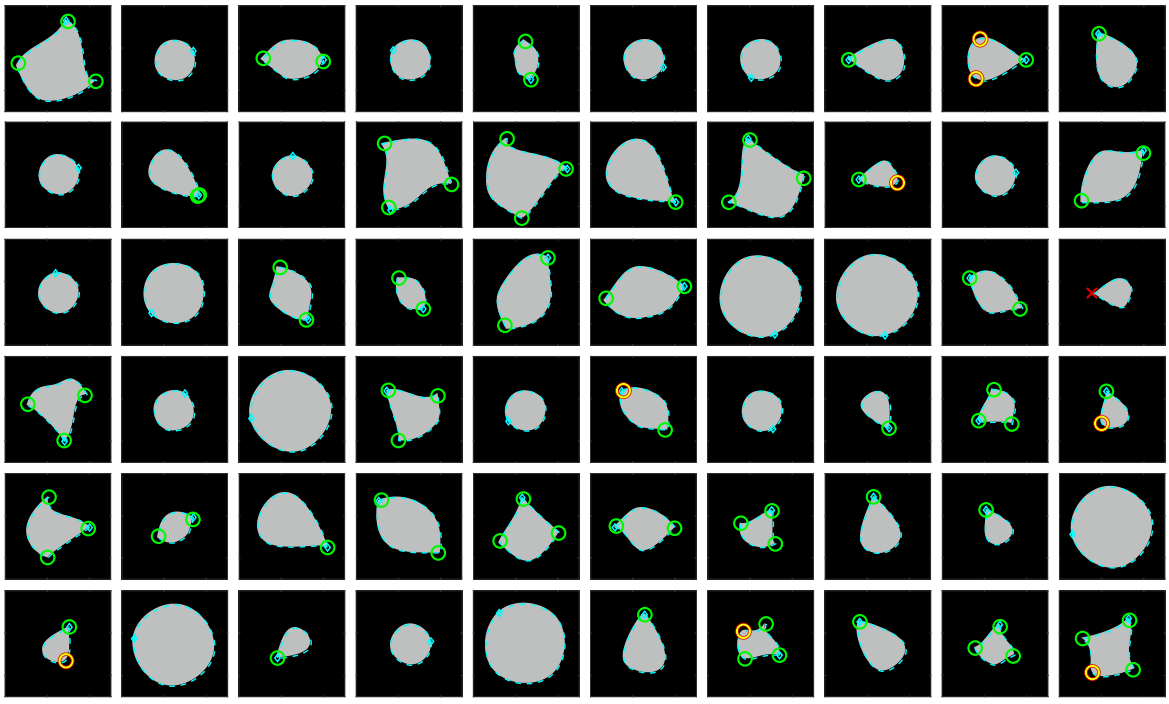


Figure 4.18 A random sample of the synthetic data used to test the peak detection using *anglegram*. In all cases, the detected junctions are presented in a green marker ($\circ$) and the ground truth junctions are shown in a red marker ($+$). Cases where a detection was made that did not have a junction are green circle markers with a red outline.

Table 4.2 Synthetic test of peak detection from 1000 generated shapes. Rows show the shape types and columns show the number of peaks detected by the algorithm. The accuracy of the method is highlighted.

Shapes / Peaks	Peaks detected (%)				
	None	1	2	3	≥ 4
Circle	**89.35**	0	0	0.47	10.18
Drop	4.42	**86.02**	6.81	2.52	0.23
Bidrop	0	0.41	**95.17**	4.31	0.11
Tridrop	0	0.02	2.76	**92.94**	4.28

Additional anglegram experiments

Preliminary measurements of elongation. As in Chapter 2, particularly in figure 4.19, repeated here, the cells' elongation and shape changes can be observed as they move. This is referred to as *pointiness*. It is noticeable how the elongation becomes greater making the peaks appear more prominent as they get closer with another cell. In this experiment with synthetic data, simulated elongation of each of the basic shapes is presented. In Figure 4.20 the difference between the lowest and highest angles of the minimum intensity projection are shown as a preliminary to the measurement of pointiness of a shape.

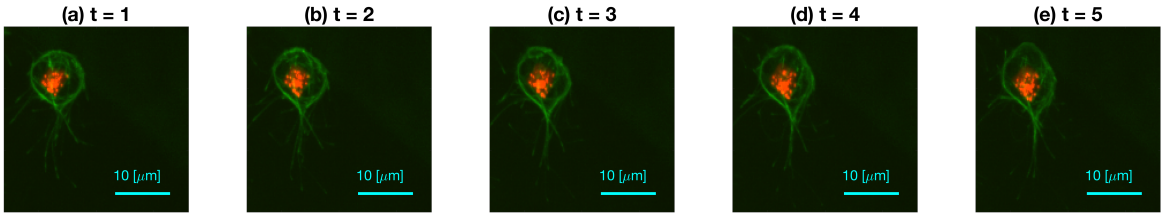


Figure 4.19 *Pointiness* changes observed in consecutive frames (repeated Figure 2.23 for convenience). The detail of the cell is shown where an array of microtubules appears to form at the bottom of the cell as it migrates. Outside the frame, another cell would approach causing the elongation.

Comparison against Harris corner detector. Harris and Stephens [103] proposed a junction detection algorithm centred in searching for extrema in the curvature of the image gradient, which is widely used. Note that, unlike this work, Harris calls all junctions corners. A comparison against this method is provided, highlighting the differences in approach from the detection of junctions with the *anglegram*.

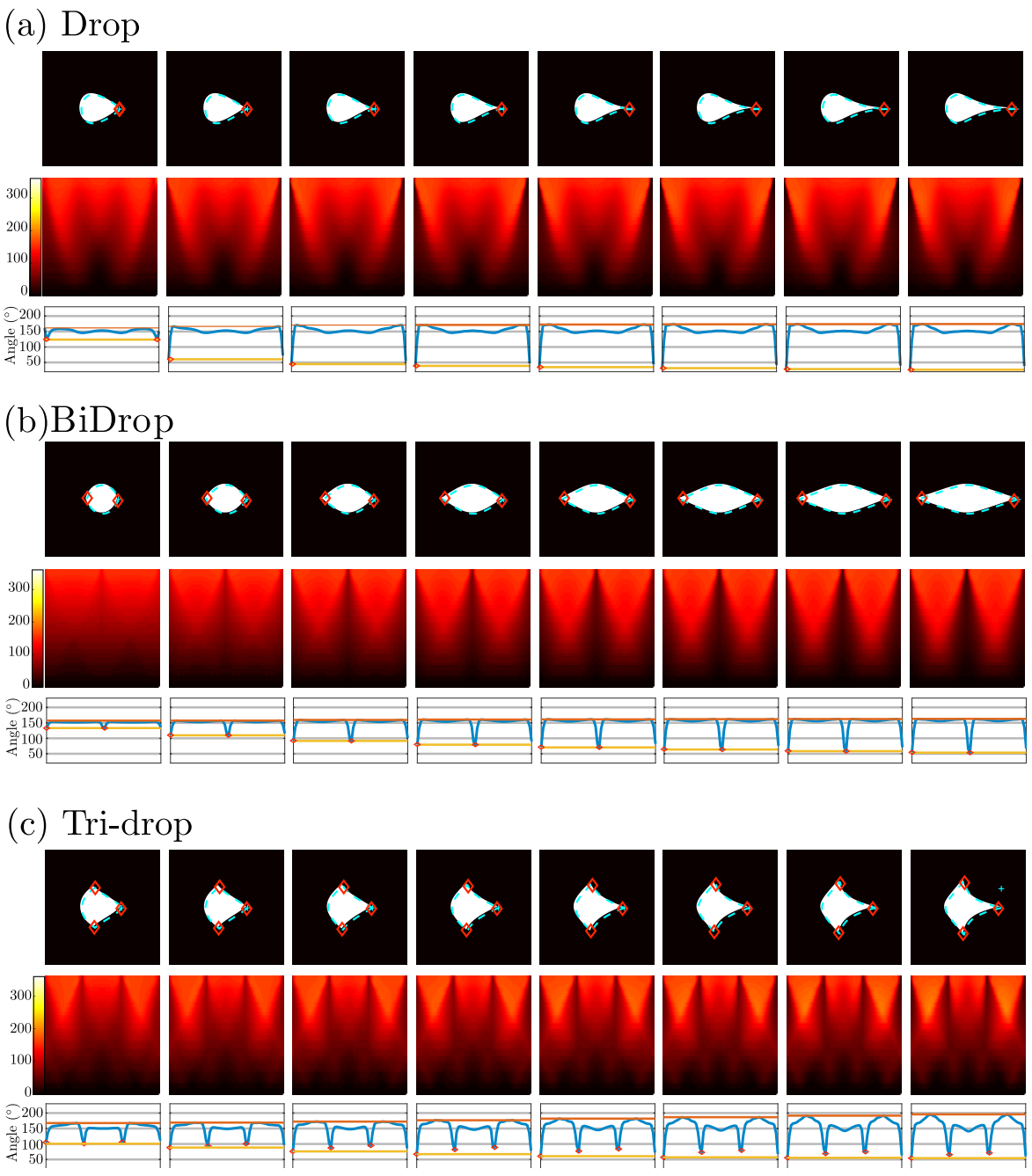


Figure 4.20 Assessment of *pointiness* of the basic shapes. Each synthetic shape was created with a visibly increasing elongation, causing the peaks to be more prominent. Eight levels of *pointiness* are presented per shape; the relationship of the anglegram values and the pointiness is presented. Notice that in all cases the difference between the maximum and the minimum values of the minimum intensity projection appear to grow proportionally to the pointiness.

In the experiment, junctions in four overlapping clumps and four basic shapes are shown, comparing the output of the *anglegram* method with the Harris corner detector. The Harris detector is ranks the junctions detected in function of its sharpness, so only the strongest corners are shown. The qualitative comparison, Figure 4.21, shows similarities in the junctions detected and missed from both algorithms and highlights the specific nature of the *anglegram* approach. Using the *anglegram* allows for a specific type of junction to be detected, and ignores other types by the definition.

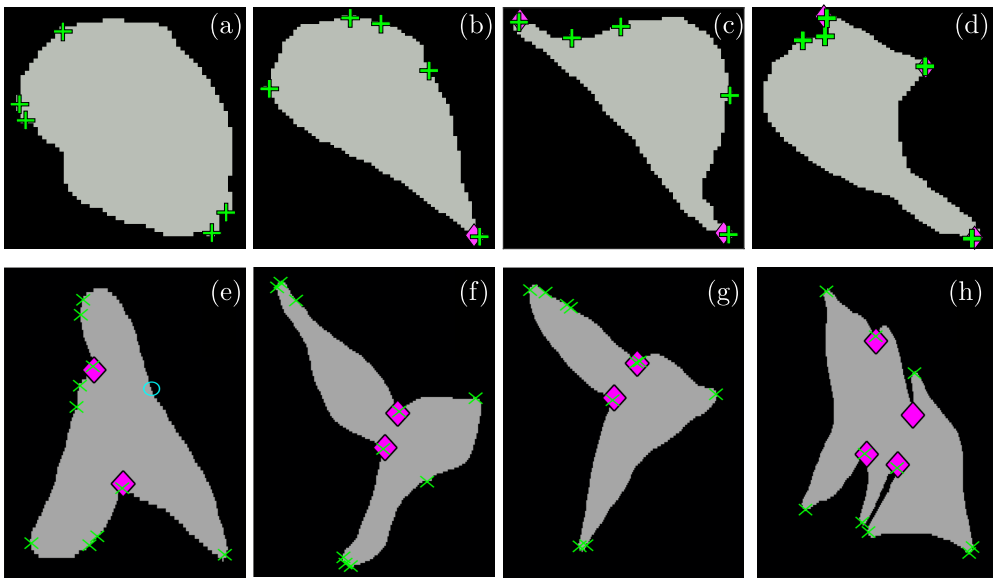


Figure 4.21 The strongest junctions detected by the Harris detector (green +) were plotted. The output of the anglegram is presented in magenta markers (⋄). The top row, (a-d) show the four basic shapes in which peaks are detected. Notice that the detection via *anglegram* shows precisely the corners of interest. The bottom row, (e-h) show the detection of corners, where only in the clump in (a) shows a missing junction from both algorithms (cyan ○).

4.6 Segmentation of overlapping objects in a single frame using the Anglegram

The following section is reproduced verbatim from [72]. Comparison of methodologies to segment clumps into overlapping cells is described. As a benchmark, Voronoi partition was used with the method developed in Chapter 3. Then, the three methods, which incorporate the information from the junctions into a segmentation output were used. The methods differed in the way the junctions' information was incorporated into a complete segmentation. Junction Slicing (JS) and Edge Following (EF) involved the

explicit use of the junctions' position, while the proposed self-organising map (SOM) fitting involved the information of junctions into creating a custom SOM that adapts to the overlapping section of the data. In this work, only the cases where two junctions were found were examined in detail. A diagram showing all methods presented and the data flow is presented in Figure 4.24.

Voronoi partition This method was included as a lower-bound benchmark for comparisons against with the proposed methods. It consists of a naïve approach to the problem that does not include any information from the green channel. The image area was partitioned, using Voronoi tessellations [94]. The partition of the green channel's cells was based on the centroids of the detected nuclei from the image's red channel. Full description of this method is presented in chapter 3.

Junction Slicing (JS) This method partitioned the clump with the line that joined two junctions. For each junction detected, each of the two adjacent segments of the boundary of the clump would correspond to one of the different objects within the clump. Since the points in the boundary are ordered, starting at one point $\mathbf{p}_1$ and moving alongside $\mathcal{B}$ in a clockwise manner, then the segment that appeared before a detected junction would correspond to one cell, whereas the segment that appeared after the junction would correspond to the other cell. As described before, Figure 4.5, for cases where only two junctions were found, the problem of selecting which pair of junctions will be joined becomes trivial. However, considering a case like the one presented in Figure 4.5, or in real data in figure 4.17, where four junctions would appear, different combinations of the boundary segments could yield different *candidates* of segments.

Edge Following (EF) In order to obtain the edge information, the Canny [79] algorithm was used on the green channel of the image. The algorithm consists of finding the local maxima of the image gradient. In this work, the parameter of the standard deviation was set to $\sigma = 1$. The trend of the two adjacent segments leading to the junction was defined by approximating the tangent line of the boundary at the junction point. The definition of the tangent line was taking an average slope of the secant lines leading up to the detected junction. The tangent line was extended, and a region of interest (ROI) was defined by a triangle where the approximated tangent line goes along the vertex and the adjacent angle corresponds to 20 degrees to each

side of the tangent line (Figure 4.24). The ROI defined for each of the adjacent line segments was then intersected with the edge information of the image, resulting in a set of binary line segments, which were labelled. Labelling of the binary line segments allowed for individual analysis of each line. Each line detected was analysed in terms of its orientation and size, preserving the one that has the most similar orientation to the extended line segment. Binary line segments with a change in direction were split by removing the strongest corners, detected using Harris' corner detector. The lines found by both ROIs on each junction were used as new coordinates to add to the boundary of the corresponding cell.

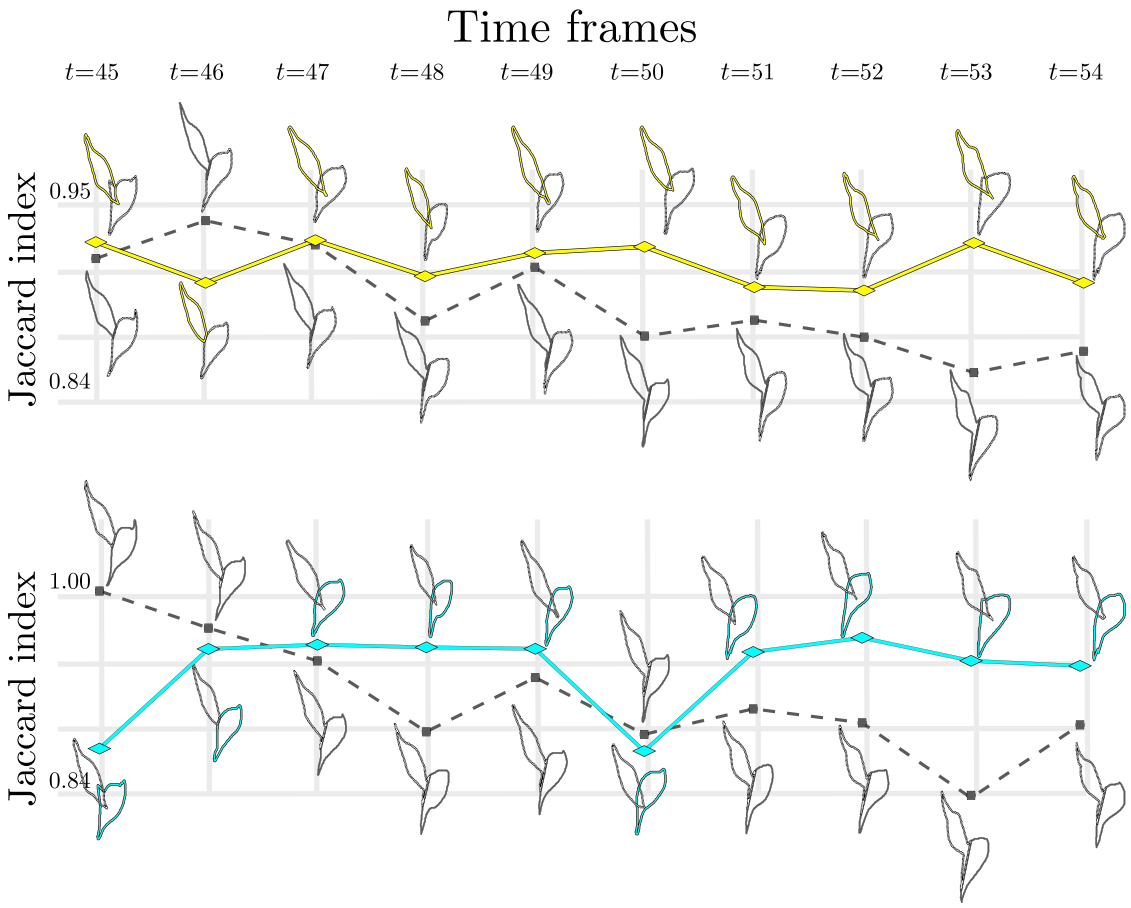


Figure 4.22 Initial study comparing Jaccard indexes of the segmentation via Edge Following versus Voronoi tessellations. The results from the initial EF segmentation are shown in the **top row** while the Voronoi output are shown in the **bottom row**. The boundaries of the corresponding cells are shown in cyan and yellow depending the cell evaluated. The boundaries are shown alongside the plot of Jaccard similarity index per cell/method.

It is worth noting that originally, the only method developed was EF, where an initial study on it was made. Figures 4.23 and 4.22 show the main output from such exploration comparing the results of the two methodologies, performing EF in two cells overlapped along ten consecutive frames.

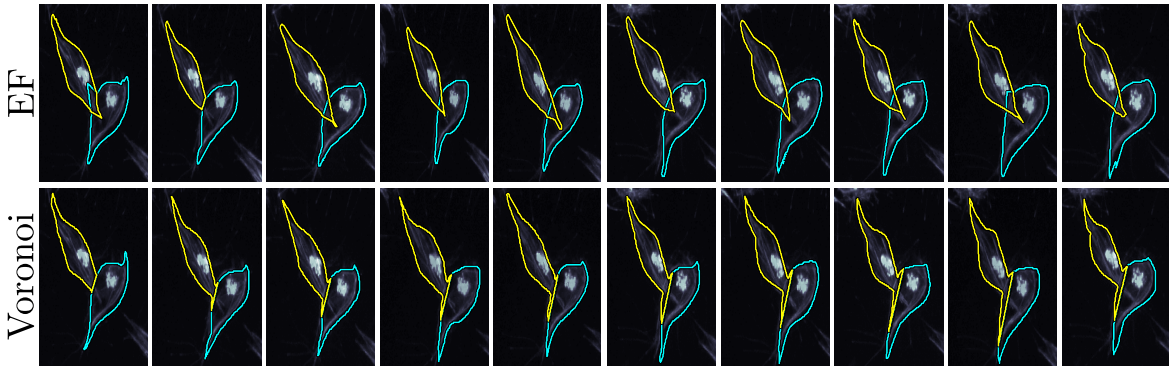


Figure 4.23 Initial study comparing segmentation via Edge Following (EF) and Voronoi tessellations. Two cells were assessed in this preliminary study, spanning ten frames, from $t = 45$ to $t = 54$. The results from the initial EF segmentation are shown in the **top row** while the Voronoi output are shown in the **bottom row**. The boundaries of the corresponding cells are shown in cyan and yellow lines.

Self-organising Maps (SOM) Fitting This work proposes an alternative implementation of the self-organising maps [105] that adapts itself to the overlapped area. For this SOM, a custom network was defined, as well as the input data and additional rules to the definition of the step-size parameter, α. Let the network $\mathcal{R} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V} = \{\mathbf{m}_i = (x_i, y_i) \in \mathbb{R}^2 : i = 1, \cdots, n_v\}$ are nodes assigned to positions in the plane and $\mathcal{L}$ are some edges linking the some of the nodes in $\mathcal{V}$. For implementation purposes, each node $\mathbf{m}_i \in \mathcal{V}$ has an identifier, position, and a *speed* parameter, related to the movement of each node. The input data was determined by the positions and normalised intensity values of the image, i.e. $(\mathbf{x}_t, \mathbb{I}(\mathbf{x}_t))$. Values in $\mathbb{I}(\mathbf{x}_t)$which were selected by an Otsu's threshold [106], and were located within a bounding box that contains the junctions. Given an input, the algorithm proposed by Kohonen [105] follows two basic steps: identifying the closest node in the network to the input, shown in Equation (4.3), and update the positions of the nodes inside a neighbourhood, determined by a distance n_e to the winner node $\mathbf{m}_c$, (4.4),

$$\mathbf{m}_c(t) = \underset{c \in \{1, \cdots, n_v\} S}{\arg\min} \|\mathbf{x_t} - \mathbf{m}_i(t)\|_2^2 \tag{4.3}$$

$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + \alpha_t \left(\mathbf{x}_t - \mathbf{m}_c(t)\right) & , (i,c) \in \mathcal{L} \text{ and } \mathrm{Dist}(\mathbf{m}_i, \mathbf{m}_c) \leq n_e \\ \mathbf{m}_i(t) & , \text{otherwise} \end{cases}, \quad (4.4)$$

where $\mathrm{Dist}(\mathbf{m}_i, \mathbf{m}_j)$ refers to the distance from node i to node j in the shortest path determined by the edges $\mathcal{L}$. In this work, the parameter α_t was determined the intensity level of the image, I, and the speed parameter of the node. The proposed formula for the parameter α_t is shown in equation (4.5),

$$\alpha_{t,i} = \alpha_0 \times \left(0.2 + I(\mathbf{x}_t)\right)^4 \times \mathrm{speed}(\mathbf{m}_i), \quad (4.5)$$

where $\mathrm{speed}(\mathbf{m}_i)$ is 0.1, or 1, depending on where the node resides in the topology. The network was defined by taking a subset of the boundary points in $\mathcal{B}$ in a ring topology, and then adding two networks in a grid topology to each side of the line joining two junctions. The three networks are independent from each other. Thus, $\mathrm{speed}(\mathbf{m}_i) = 0.1$, if $\mathbf{m}_i$ was located in the boundary of the clump, and $\mathrm{speed}(\mathbf{m}_i) = 1$, if it was one of the grid networks. The assumption is that the network taken from $\mathcal{B}$ would be closer to the actual cell, and therefore it should not move abruptly, whereas the networks inside the clump will adjust and adapt to the shape of the overlapping area between the cells. In order to finalise the network final state into a segmentation, the external network was taken as a new clump and it was partitioned by the same line used in the junction slicing (JS) method. Finally, the area formed by the inner network that adapted to the overlapping section of the cell was dilated with a 5×5 square element and then attached to both partitions of the new clump. The right column of Figure 4.24 displays the main steps of the SOM fitting method described.

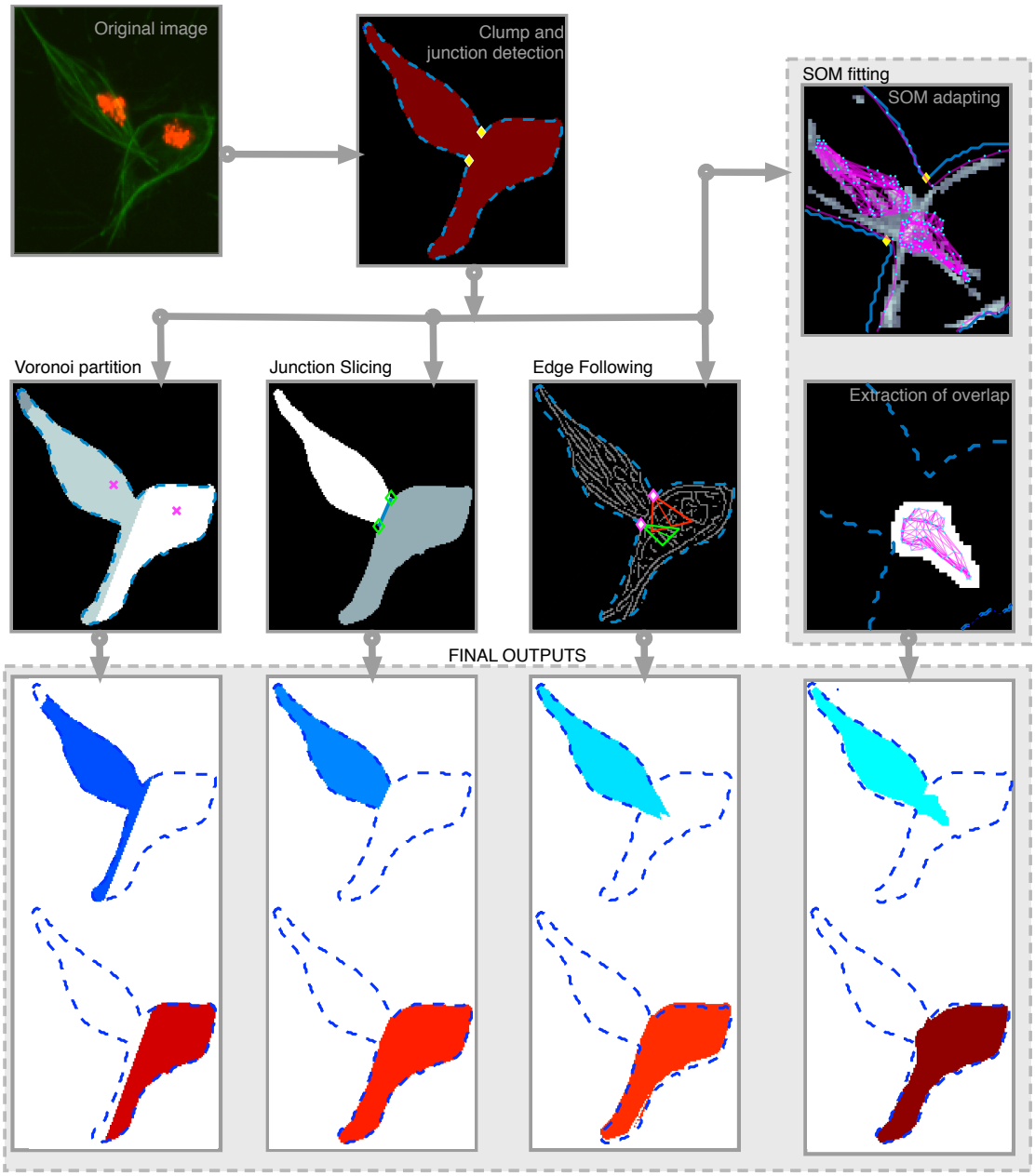


Figure 4.24 Illustration of all the methods developed and the workflow to obtain results. Top left shows the detail of CLUMP (a) in the original frame. Clumps are detected and the boundary was extracted. With the boundary information, the *anglegram* was calculated and the junctions were detected (top, middle). On the second row, a diagram to the methods were presented. From left to right, the Voronoi partition, Junction Slicing (JS), Edge Following (EF) and SOM fitting. Bottom row shows the outputs from each method for both cells within the detected clump.

Overlapping segmentation results and discussion

The following section is reproduced verbatim from [72]. The author's work presented in [71, 72] shows the initial approach of solving the ambiguity introduced by the baseline segmentation of the green channel in the data, forming clumps. Two clumps were assessed for this experiment spanning ten consecutive time frames with available ground truth. A qualitative result is presented in Figure 4.25, where the four techniques can be assessed in one representative frame. The detail of both clumps is shown with the boundary of the segmentations superimposed.

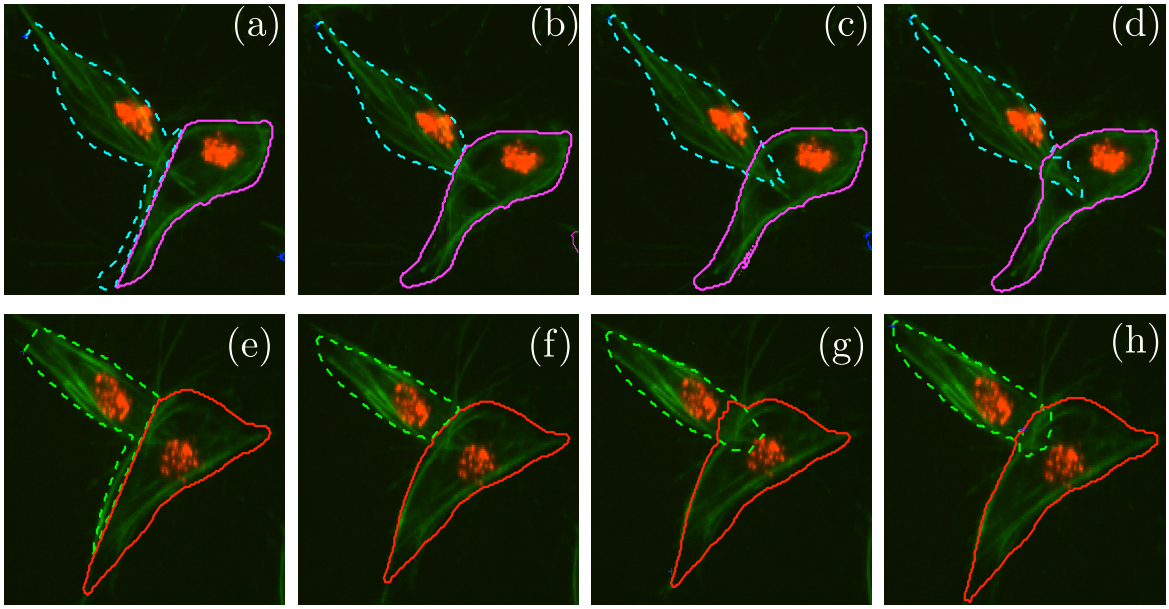


Figure 4.25 Qualitative comparison of different segmentation methods in two different clumps. Two clumps are shown, one in (a-d) and the other one in (e-h). The columns have the results from the frame depending on the different techniques described: (a,e) Voronoi, (b,f) JS, (c,g) EF and (d,h) SOM. Image rearranged from [72].

The data was organised in the following way: two clumps, with two cells each over ten frames. Furthermore, the comparison with the ground truth was performed with three performance measurements: Precision, Recall and Jaccard index. Thus, 120 calculations were made. To summarise them, tables 4.3 and 4.4 were created. Junction Slicing (JS), Edge Following (EF) and Self-organising Maps (SOM). The Jaccard Similarity Index [73], recall and precision [74] statistics were computed for both clumps on all the frames and all the methods described, box plots of the results are shown in Figure 4.26 and summarised in table 4.3. Table 4.4 presents a statistical analysis of the results presented in Table 4.3. The Wilcoxon Signed Rank test [107] was implemented

to compare the medians of the results per measurement, for example Precision(Voronoi) vs. Precision(JS). The test was chosen as an alternative to a traditional t-test, without the need to assume the distribution of the measured values.

Table 4.3 Comparison of mean values of Precision, Recall and Jaccard Index for clumps 2 and 3 over 10 frames. This table summarises the results in Figure 4.25. Highest results are highlighted.

	CLUMP (a)			CLUMP (b)		
	Precision	Recall	Jaccard Index	Precision	Recall	Jaccard Index
Voronoi	0.906	0.925	0.843	0.872	0.868	0.771
JS	**0.970**	0.953	0.926	**0.974**	**0.948**	**0.925**
EF	0.964	**0.983**	**0.948**	0.938	0.950	0.896
SOM	0.965	0.951	0.919	0.973	**0.948**	0.923

Table 4.4 Statistical analysis of results presented in table 4.3. The Wilcoxon Signed Rank test [107] was implemented to compare the results per measurement (Voronoi, JS, EF and SOM). The table presents the p-values on the paired test for each of the pairs, (first and second columns). Tests where the null hypothesis could not be rejected are highlighted.

		CLUMP (a)		
		Precision	Recall	Jaccard Index
Voronoi vs.	JS	$p = 0.002$	$p = 0.002$	$p = 0.002$
	EF	$p = 0.004$	$p = 0.004$	$p = 0.004$
	SOM	$p = 0.002$	$p = 0.037$	$p = 0.002$
JS vs.	EF	$p = 0.004$	$p = 0.004$	$p = 0.004$
	SOM	$p > 0.05$	$p > 0.05$	$p > 0.05$
EF vs.	SOM	$p > 0.05$	$p = 0.004$	$p = 0.008$
		CLUMP (b)		
		Precision	Recall	Jaccard Index
Voronoi vs.	JS	$p = 0.004$	$p = 0.004$	$p = 0.004$
	EF	$p = 0.008$	$p = 0.008$	$p = 0.008$
	SOM	$p = 0.004$	$p = 0.004$	$p = 0.004$
JS vs.	EF	$p = 0.008$	$p > 0.05$	$p > 0.05$
	SOM	$p > 0.05$	$p > 0.05$	$p > 0.05$
EF vs.	SOM	$p = 0.016$	$p > 0.05$	$p > 0.05$

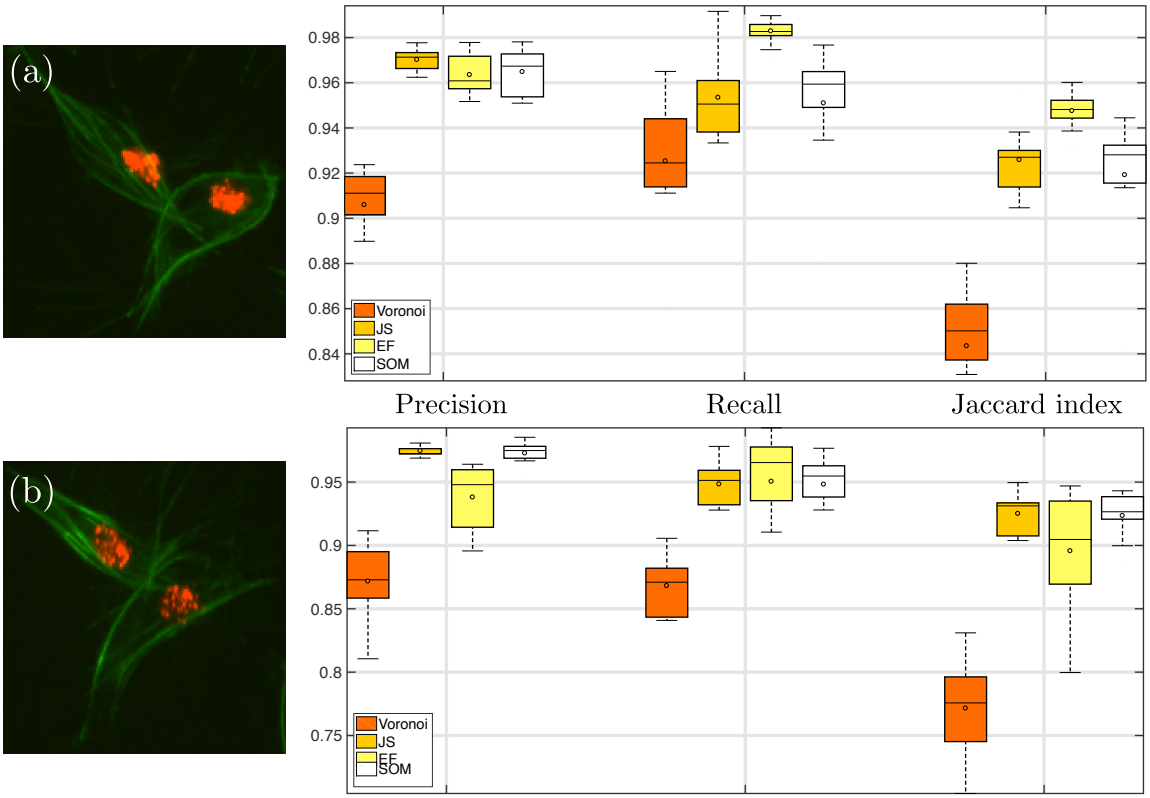


Figure 4.26 Comparison of Precision, Recall and Jaccard index for all segmentation methods presented in this section. The horizontal axis shows the different measurements of performance tested on the different overlapping segmentation methods: Precision, Recall and Jaccard index. Groups of four boxplots are shown as a visual representation of the statistical distribution of the values of each method. From left to right, the order of the methods shown are Voronoi, JS, EF and SOM. Image rearranged from [72].

4.7 Summary

This chapter presented a tool for the detection of junctions and the analysis of the shape of cells. The baseline segmentation for all developments in this and next chapter was introduced. As mentioned in section 3.4, more sophisticated approaches would produce unsatisfactory results, which failed to solve the ambiguity of the clumps. The **baseline segmentation** is a fast and robust algorithm, with relatively low issues. As observed in Figure 4.4, some limitations were observed in the formation of clumps, as the microtubules in the cells cause larger clumps. This effect is likely due to the morphological hole filling after thresholding. However, such a trade-off is necessary to allow for boundaries to be closed and *almost* convex, as defined in Section 2.3.

The ***anglegram*** matrix was introduced, which paired with the projections along its rows, provides a signature of the shape of an object. The anglegram allows for the correct identification of junctions when considering a *clump* or a single cell. Tables 4.1 and 4.2 show promising results for both cases. The method can be exploited in several ways, from allowing the reconstruction of an overlapped segmentation, to allowing a measurement of the elongation of the cells. This chapter also shows some limitations to the method, and poses some restrictions on the underlying segmentation, like the production of a closed boundary. Regarding the limitations, when detecting corners, the anglegram method was unable to detect anything in a range of low angles, $[188 - 192]$ in the experiments presented, although the Harris corner detector showed similar problems at those angles, Figure 4.21. In peak detection, the limitations lie in the amount of corners found. Notice in Table 4.2 that for all shapes with at least a peak, the detection of an extra peak is more common than missing peaks. Some limitations are shared with the Harris corner detector, notice in figure 4.21(e) a junction was missed by both methods.

The information that can be extracted from the anglegram shows a good promise for shape determination and analysis, whether this involves overlapping or non-overlapping objects. The limitations for corner detection depends on the participating cell positions, and it is transferred to the following segmentation methods. Table 4.3 shows a better performance from all three junction-based methods compared to the Voronoi partition. Furthermore, the percentile box sizes in figure 4.26 show that the EF method (yellow) is less consistent than the SOM method (white).

This chapter involved the analysis of single frames, however it is important to remember that most of the information of interests lies within the movement of the cells. In the work by Stramer et al. [4], the interactions amongst the cells' structures appear to anticipate migration, where sharp corners suggest an active migrating cell and rounded corners inactivity. Different shapes of the could hint at different movements. Therefore, a classification of the cell shapes into distinct shape-states could aid in biological studies where a precise analysis of cell shapes as they evolve through time is necessary. Such methodologies will be explored in the following chapter.

Chapter 5

A Cell Shape Analysis and Tracking Algorithm: `macrosight`

In this chapter, the algorithms to study macrophages cells are described. Combined with the previous chapter, the algorithms in this chapter belong to a framework which allows for different types of analysis. The previous chapter presented the anglegram and some methods allowing for single frame analysis, without incorporating the temporal context. The current chapter presents **two distinct contributions** from the ones presented in the previous chapter; both analyses were performed on the macrophages datasets. The temporal context is included in both by the tracking of the cells. For tracking, the keyhole algorithm within PhagoSight [102] is used. The methods proposed in this chapter, in conjunction named `macrosight`, exploit the baseline segmentation from Chapter 4, using the channels for different purposes: (i) the red channel is used for tracking of the cells and (ii) the green channel is used to perform the different analyses.

An overview of the `macrosight` framework is described below. Figure 5.1 shows a graphical abstract in which the two projects and its relationship with the baseline segmentation, PhagoSight and how both techniques were incorporated into the two main projects is described. As discussed in Chapter 3, PhagoSight is a framework which includes segmentation and tracking routines, the latter implementing the keyhole algorithm. Apart from the algorithms, information is handled and saved in a specific way. To incorporate the baseline segmentation output into PhagoSight's framework, technical modifications were made to the output and storage of the segmentation and to the output of the framework. Furthermore, information regarding the clump was incorporated. Section 5.1 is included as a detailed overview of PhagoSight and the

modifications made to it, allowing the implementation of developments in sections 5.2 and 5.3.

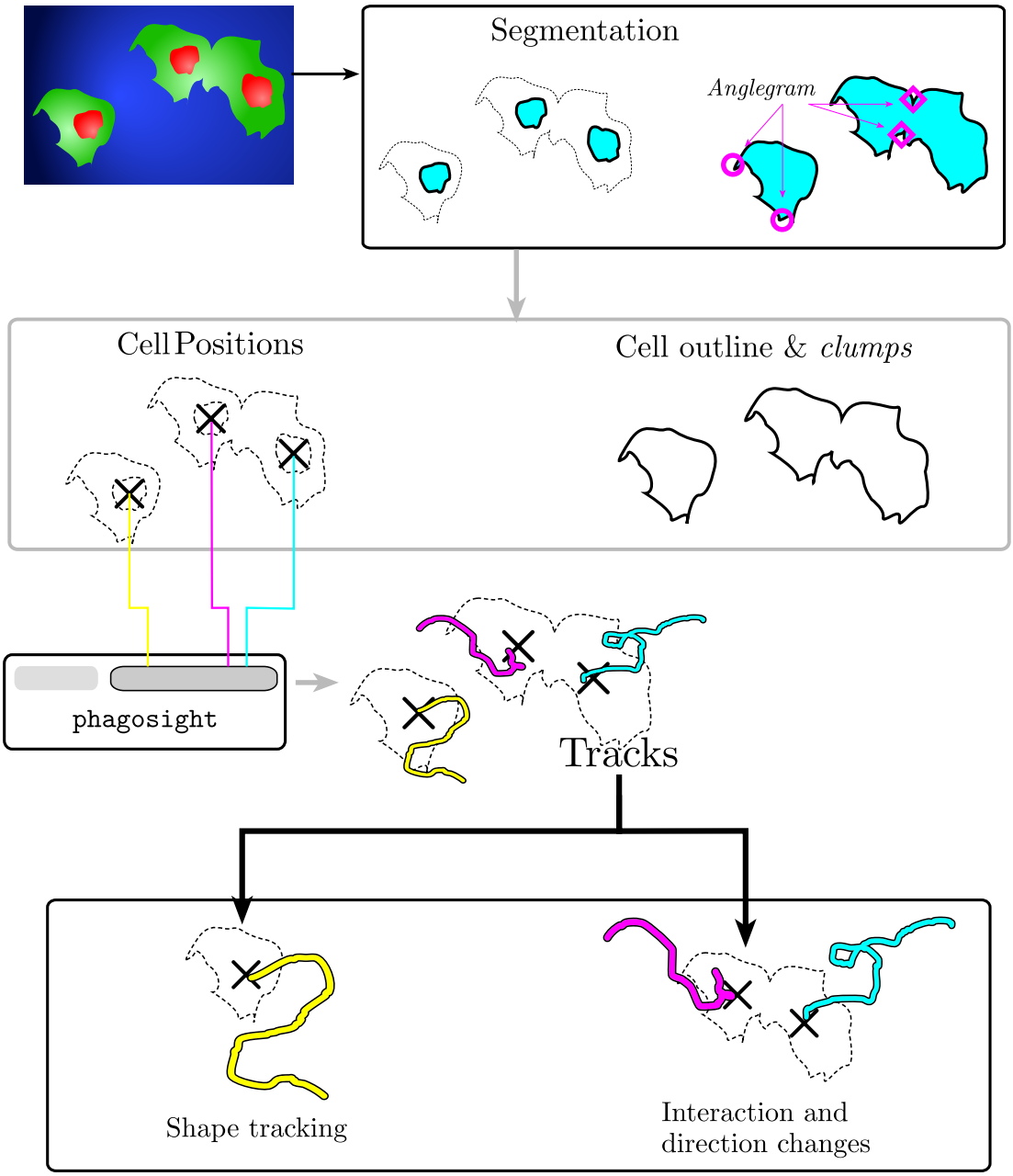


Figure 5.1 Complete graphical abstract of `macrosight`, which includes PhagoSight, the anglegram, shape analysis and analysis of cell-cell interactions. The different contributions are clearly presented. At the top the segmentation block is shown, including the **baseline segmentation** and the **anglegram**. At the bottom, the two contributions are shown: **shape tracking** and **direction changes** from cell-cell contact.

The first project, shown in section 5.2, consists of the incorporation of shape analysis to a tracking framework, and was originally presented in [108]. Macrophages cannot be assumed to have simple shapes, however from one time frame to the next, the shape does not change drastically, but rather evolves gradually. The project describes an algorithm that iteratively tracks and analyses the shape of macrophages in time intervals when they do not overlap. The process allows the observation of shape variations as each cell migrates and the measurement of shape parameters. The *anglegram* is used to detect peaks in the cells as they evolve. The second project, shown in section 5.3 presented in [109], where the direction of the original path changes from at least one cell. The work analyses the change of direction in cells before and after overlapping. Interaction of macrophages are assumed to occur upon coming in contact, which causes their representation in the green channel to overlap.

Notation. The notation relevant this chapter include the time position represented with the letter t, $t+1$ refer to the following frame in the dataset and $t-1$ represents the previous frame. In section 5.3, specific frames are referred to via subindexes. The positions in the image are represented in vector notation $\mathbf{x}_t$ or explicitly (x_t, y_t). Tracks are represented by $\mathcal{T}_i$, where i is an index and represent collections of positions of the cell as it moves in time. Cell boundaries are represented by $\mathcal{B}$, as in chapter 4.

5.1 Modification of the PhagoSight framework for migrating macrophages

This section describes the technical details which were considered when incorporating the macrophages time sequences into PhagoSight and use the keyhole tracking algorithm. PhagoSight is a semi-automatic software for the segmentation and tracking of fluorescent neutrophils, it has a complete pipeline (algorithm 1) based on saving the intermediate steps in the processing. Storing intermediate steps of the processing is visible in lines **4** and **9** of algorithm 1. The software recognises the dataset containing the raw data, DS, and produces intermediate directories DS_{Re} for preprocessed data and DS_{La} for the segmented and labelled images. Therefore, to incorporate a different segmentation technique, the right information has to be saved to the correct directory and naming scheme. To incorporate the results of a different segmentation technique, the segmented data must be labelled and stored in a directory with the name DS_{La}. The software has

the ability to recognise which of the intermediate steps has been previously detected and continue the analysis from there.

The PhagoSight package was used to track the movement of the nuclei through time using the keyhole algorithms. Its segmentation is similar to the baseline segmentation but only uses scaling of a single Otsu threshold to produce the two different levels. Also, it does not account for the different channels in the data, for which the channels were saved as different images. The output of PhagoSight is the `handles` structure, which contains the general information of each image analysed and the information of the cells per frame. Namely, which of the detected cells in frame t is parent to another cell in $t+1$. The information stored per track detected is in table 5.1. The information used in the subsequent analyses are the position in time and space and two labels associated to each cell.

Algorithm 1: PhagoSight pipeline

Input: Dataset location: DS, Input parameters: θ (optional)
Output: Structure variable: `handles`
foreach *image* $\mathbb{I}$ *of the dataset* DS **do**
  $\mathbb{I}_{Re} = preprocess(\mathbb{I})$
end foreach
Save preprocessed images Re to disk: DS_{Re}
Determine parameters: θ
foreach *image* $\mathbb{I}_{Re}$ *of the dataset* DS_{Re} **do**
  $\mathbb{I}_{La} = segment(\mathbb{I}_{Re})$
end foreach
Save labelled and segmented images $\mathbb{I}_{La}$ to disk: DS_{La}
Perform **tracking** on labelled images DS_{La}
`handles`= structure with locations of DS, DS_{Re}, DS_{La} and tracks information.

First, a **label in binary image**, which refers to the label assigned to the cell at segmentation and the **final track label** corresponds to the track identification. Thus, a single cell will have a **final track label**, or track label, identifying it in the context of the whole experiment and at each frame it will have a **label in binary image** which could vary from frame to frame.

Table 5.1 Track information retrieved from PhagoSight framework (repeated from Table 3.4 for convenience). The information utilised for the analysis is highlighted: **position** provides the coordinates x_t, y_t, z_t of the cell in the image per frame t, **time frame** is the temporal position in the dataset, t, **label in binary image** refers to the label assigned to the cell at segmentation and **final track label** corresponds to the track identification, $\mathcal{T}$.

Position x_t, y_t, z_t	Distance to closest	**Time frame**
ID	Parent cell	Child cell
Velocity	Volume	**Label in binary image**
Keyhole	Track ID	**Final track label**
Bounding box information	Volume to surface	Sphericity
Neighbours at different distance brackets	Distance to disappear	Distance to appear

Addition of clump information. Each track produced by PhagoSight includes the information of the cell's nuclei that has been tracked and linked from one frame to others. The information of interest to this work is explained in Table 5.2, it includes parameters such as time frame, position and velocity of each nuclei at each time frame. At each point in time, the presence of clumps was detected by counting the number of nuclei contained within a single object detected in the green channel.

Table 5.2 Brief description of the parameters measured per track.

Field	Description
Time frame t	Frame in the dataset.
Position (x_t, y_t)	Cartesian coordinates of the centroid of the detected nucleus at time t.
Track label $\mathcal{T}$	Unique identifier for each track.

As each nuclei within a clump has a track associated to them, each clump can be uniquely identified via a simple *unique identifier number* or for short *code*, which includes the labels of the tracks contained within it. For instance, let r, q be the labels of two tracks ($r < q$) which at a certain point in time belong to a *clump*, then the code c is defined by $c(r, q) = c(q, r) = 1000q + r$. The value of 1000 is chosen arbitrarily as a large number, larger than the number of total tracks. Notice how the tracks' labels are arranged from left to right starting with the highest identifier to the lowest; for example, code `24013` would correspond to a clump that at a certain frame contains tracks 24 and 13. The previous definition can be extended for an arbitrary number

of labels m interacting in the same clump as $c(r_1, \cdots, r_i, \cdots, r_m) = \sum_{i=1}^{m} 1000^{i-1} r_i$. Each clump can be uniquely identified based on the tracks contained in it. Table 5.3 shows a simple example of the creation of the clump codes. The inclusion of the codes facilitates the analysis of the cells that interact with each other.

Table 5.3 Examples of clump codes created through the track labels obtained by PhagoSight by applying the defined codes. See text for detailed explanation.

Clump code	Code construction $c(\cdot)$	Tracks within *clump*
`2001`	$c(2,1) = 2000 + 1$	2,1
`3002`	$c(3,2) = 3000 + 2$	3,2
`5003002`	$c(5,3,2) = 5000000 + 3000 + 2$	5,3,2

The inclusion of the clump information allowed to partition tracks and shed light on the state of the cell: single or involved in a clump. For example, Figure 5.2 represents the information from each track, as produced by PhagoSight, where the clump code is incorporated by `macrosight`. The information of a single track (track ID = 2) is shown, at different time frames. When the cell is not part of a clump, the variable **clump code** has a `0`. At a specific frame $t + 1$, the cells in the green channel come in contact, forming clump `2001`. At frame t_1, two more cells come in contact with clump `2001`, thus forming clump `5003002001`.

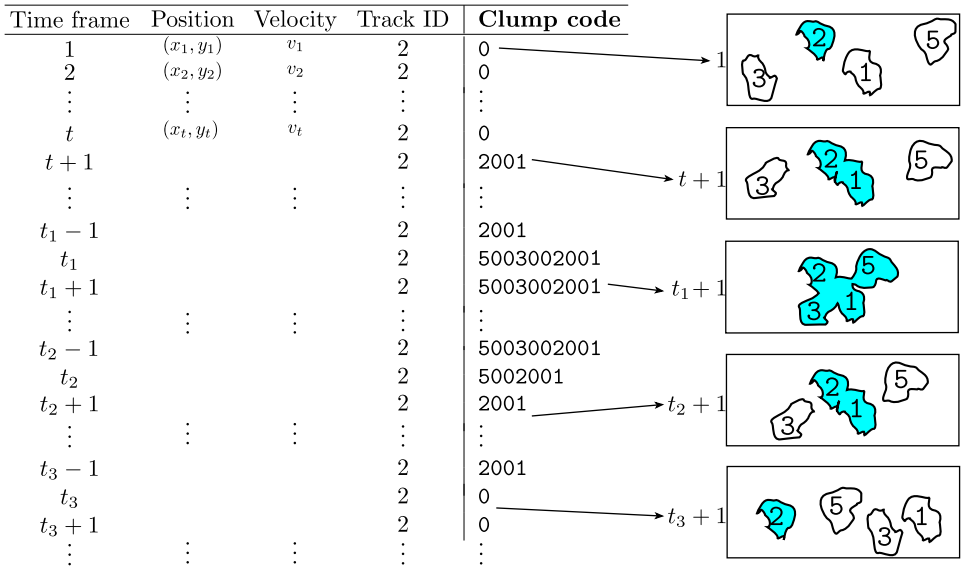


Figure 5.2 Illustration of clump codes incorporated to a particular track information. The table shows the information of track `2` spanning along certain time frames and in several clumps. The right column shows a representation of the cells at different time frames, and their involvement in different clumps.

5.2 Shape tracking

This section describes a framework within `macrosight` developed to track and analyse the shape of migrating macrophages from the baseline segmentation output, and follows the description made by the author in [108]. The work is based on the notion that the red channel can be segmented robustly, which would allow the keyhole tracker [102] to aid in the following of the cells' paths, which could be incorporated into tracking the shape of the green channel, and analyse its shape detecting corners with the *anglegram*.

Motivation. This approach may appear redundant having a segmentation which can obtain clumps and single cells in a quicker way, however, this approach allows a complete study of the cell's evolution allowing recording of measurements. Furthermore, if the overall position of a cell is resolved, then following the shape as it evolves in time could aid in producing a segmentation for overlapping cells in a single clump.

Description of the methodology

The shape of the cells was extracted from the green channel whilst the red channel was used to distinguish between individual and overlapping cells. Figure 5.3 represents an overview of the method implemented.

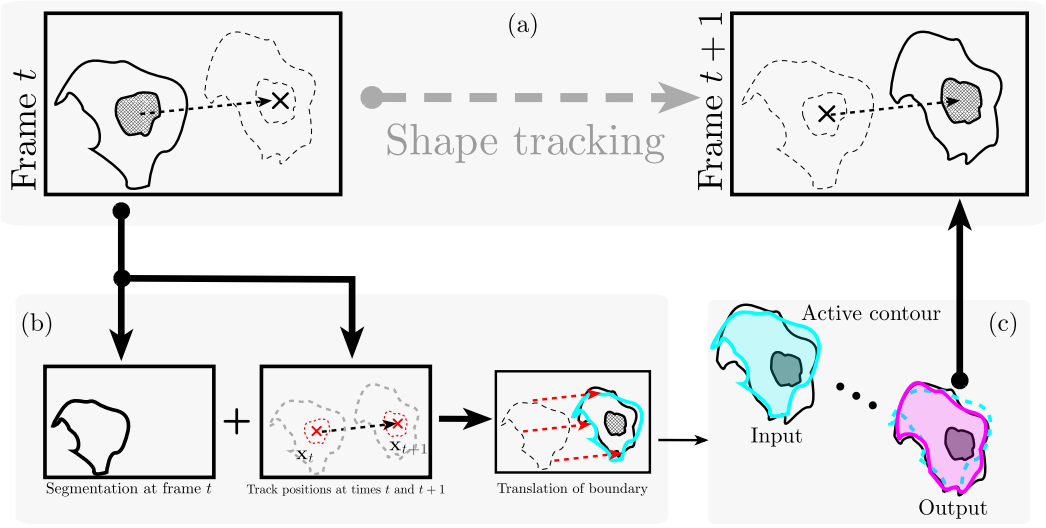


Figure 5.3 Illustration of shape tracking functionality.

The method focuses in tracking the non overlapping cells. The main functionality presented is a framework in which a cell shape can be tracked through a curve evolution

algorithm. PhagoSight [102] was used to track the movement of the nuclei through time using the keyhole algorithm, and the baseline segmentation of the green channel was used to determine which nuclei were involved in a clump. Clumps were removed from the analysis. As mentioned in section 5.1, the tracking produced unique labels for each of the tracks detected, as well as the position of each nuclei at each point in time. Each track is represented by $\mathcal{T}_j$, with $j = 1, 2, \cdots, M$ being the number of tracks. Each $\mathcal{T}_j$ contained the information regarding the positions of the centroid of the nuclei $\{\mathbf{x}_{j,k}\}_{k=1}^{T}$ as well as the time frames where the track was present $t_{j,0}, \cdots, t_{j,k}, \cdots, t_{j_{0,T-1}}$ for the T consecutive time frames. For simplicity, a time frame of an arbitrary track $\mathcal{T}$ is shown as t_k.

Given a track $\mathcal{T}$, the shape of a cell $\mathcal{B}_{k+1}$ at any time t_{k+1} can be determined by the shape of the previous frame $\mathcal{B}_k$, and the position change from t_k into t_{k+1} of the red nuclei. Let $\bar{\mathcal{B}}_k$ be the shape $\mathcal{B}_k$ when moved onto the position $\mathbf{x}_{k+1}$. The shape $\mathcal{B}_{k+1}$ will be determined by **evolving** $\bar{\mathcal{B}}_k$ the shape from the previous time frame, where $\mathbb{I}_k$ corresponds to the image of the k_{th} time frame (Algorithm 2). Every iteration of the algorithm consists of loading the known frame t_k, the position information from the unknown frame t_{k+1}, performing the evolution from $\bar{\mathcal{B}}_{k+1}$ to $\mathcal{B}_{k+1}$ and obtaining the region properties of the new cell shape, these include (i) the *orientation*, (ii) the ratio of the minor and major axes (*aspect ratio*), (iii) the *solidity*, and (iv) the *equivalent diameter*.

Algorithm 2: SHAPE EVOLUTION Tracks shape of cells in a single track.

Input: Track: $\mathcal{T}$, time frames: $(t_0 : t_{T-1})$
$t_k \leftarrow t_0$;
$(\mathbb{I}_k, \mathbf{x}_k)$load frame information at t_k;
$\mathcal{B}_k \leftarrow$ get boundary$(\mathbb{I}_k)$;
for t_{k+1} *in time frames* **do**
  $t_{k+1} \leftarrow t_k + 1$; $(\mathbb{I}_{k+1}, \mathbf{x}_{k+1}) \leftarrow$ load frame at t_{k+1}; $d_k \leftarrow \mathbf{x}_{k+1} - \mathbf{x}_k$;
  $\bar{\mathcal{B}}_{k+1} \leftarrow$ move boundary$(\mathcal{B}_k, d_K)$;
  $\mathcal{B}_{k+1} \leftarrow$ evolve$(\mathbb{I}_{k+1}, \bar{\mathcal{B}}_{k+1})$;
  save$(\mathcal{B}_{k+1}, \text{regionprops}(\mathcal{B}_{k+1}))$;
  $t_k \leftarrow t_{k+1}$;
  $\mathcal{B}_k \leftarrow \mathcal{B}_{k+1}$;
end for

The **evolve** function in Algorithm 2 implements the Chan-Vese active contour [110, 111] method in MATLAB®. The function uses $\bar{\mathcal{B}}$ as initialisation and is able to change its parameters based on one of three states: Shrink, Grow, or Normal (Table

5.4). The active contour runs once, with a set of parameters, then the area of the output is compared to the area of the input. The parameters would be adjusted to contract or expand the shape and the active contour is re-run. To avoid an excessive segmentation leaking/contraction, the area of the output was kept within ±5% of the previous frame's area.

Table 5.4 Parameters used of the active contour function based on the desired state required. The parameters were chosen empirically through numerous tests.

State	Iterations	Smooth factor	Contraction bias
Normal	50	1.5	-0.1
Shrink	100	1.25	0.10
Grow	200	1.00	-0.25

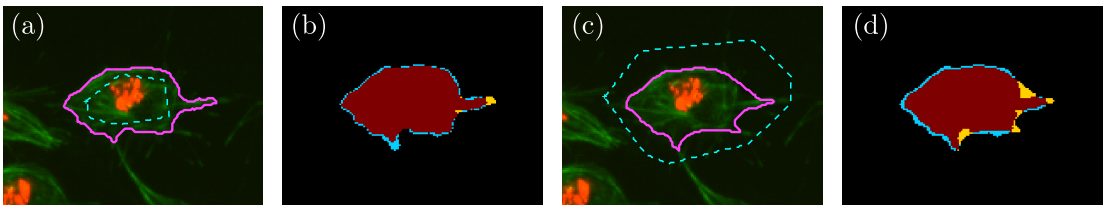


Figure 5.4 Comparison between the shrink (a,b) and grow (c,d) states of the **evolve** function. (a,c) Show the initial (cyan, dotted) and final (magenta, solid) outlines from the active contour method. (b,d) Show the comparison between the baseline segmentation and the final contour for each state.

Overlapping segmentation through shape tracking: preliminary study The approach presented in algorithm 2 could theoretically be extended beyond the scope of single cells. The tracks follow the nuclei of a cell regardless if the corresponding cell shape in the green channel undergoes overlapping with other cells. Preliminary results were not conclusive, since the implementation of the active contours would leak. However, in some cases the overlapping was resolved, especially in cases where the baseline segmentation would produce larger clumps than the corresponding ground truth showed, recall the examples presented in Figure 4.4.

Shape analysis through *Anglegram*

Described in the previous chapter, the *anglegram* [72] is a matrix which describes multiscale angle variation of a shape. The boundary of an object consists of a set of ordered points $\mathbf{p}_i$. The angles adjacent to each point are measured from the jth previous point $\mathbf{p}_{i-j}$ to the following jth position $\mathbf{p}_{i+j}$ are depicted as $\theta_{i,j} = \measuredangle \mathbf{p}_{i+j}\mathbf{p}_i\mathbf{p}_{i-j}$. This collection of angles forms the *anglegram* matrix, of which the maximum intensity projection (*MIP*) per columns is calculated. To detect clumps, the angles of interest were those larger than one standard deviation (*std*) above the mean of the MIP. In this work, a similar idea is explored, but for *acute* angles, using the minimum intensity projection (*mIP*) and the threshold is one *std* below the mean. Implementing the anglegram matrix consists of the detection of junctions that correspond to the corners of the analysed boundary. Figure 4.10 illustrates the junction detection functionality of the anglegram. The method works as reported in Section 4.6 with two alterations: (i) resizing the anglegram to have 64 rows to reduce noise, and (ii) taking the *mIP* of the first half columns, as the final columns are lower by the definition of the inner point angle measurements.

5.3 Analysis of the interactions of cells

The second analysis in `macrosight` is a method to analyse the movement of macrophages was proposed. This section follows the work presented in [109]. Particular emphasis was placed in the analysis of the cells' trajectories and the interaction between cells. The algorithm integrates the segmentation algorithm described in [72] into the software framework PhagoSight [102], and utilises the overlapped clumps as points of interaction between the cells. The main hypothesis of this work is that the direction of a given cell will change noticeably before and after an interaction with another cell. Figure 5.5 represents the hypothesis and the key experiment performed. The experiment is a comparison between the trajectories of (i) the single cells moving without noticeable external influences from other cells and (2) the path of cells before and after coming into contact with one or more cells. A preliminary approach to determine cell behaviour would involve the detection of a change in trajectory after an interaction within a clump.

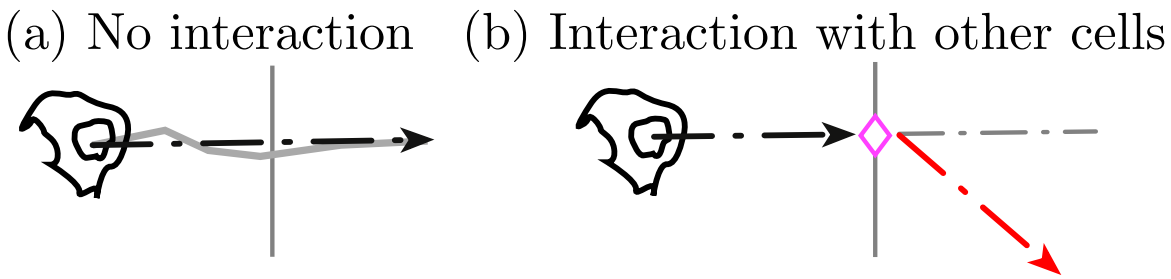


Figure 5.5 Outline of the main hypothesis in this work. The interactions of the cells in a clump appear to influence on the migration patterns of the cells. The diagram shows (a) the case where a cell's trajectory does not change significantly from a point chosen arbitrarily; (b) shows the expectation of a cell that interacting in a *clump* and changing direction noticeably.

Description of the methodology

The method in this work can be divided into four stages of analysis, referring to the framework in Figure 5.1. First, the segmentation of each channel individually. Then, the tracking of the objects detected in the red channel is performed, and the detections of each channel are classified as clumps or single cells. The crucial aspect of this part of the framework consists of an analysis of the tracks' directions before and after the clump. For each track, the change of direction (Δ Direction) is estimated before and after a given clump. Figure 5.6 shows an illustration of the procedures carried out.

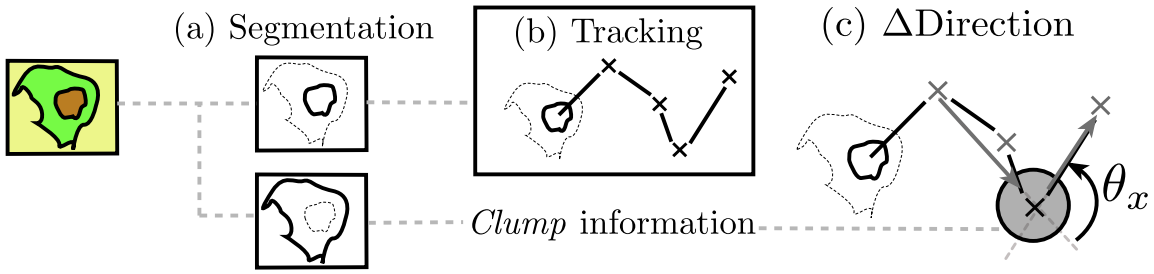
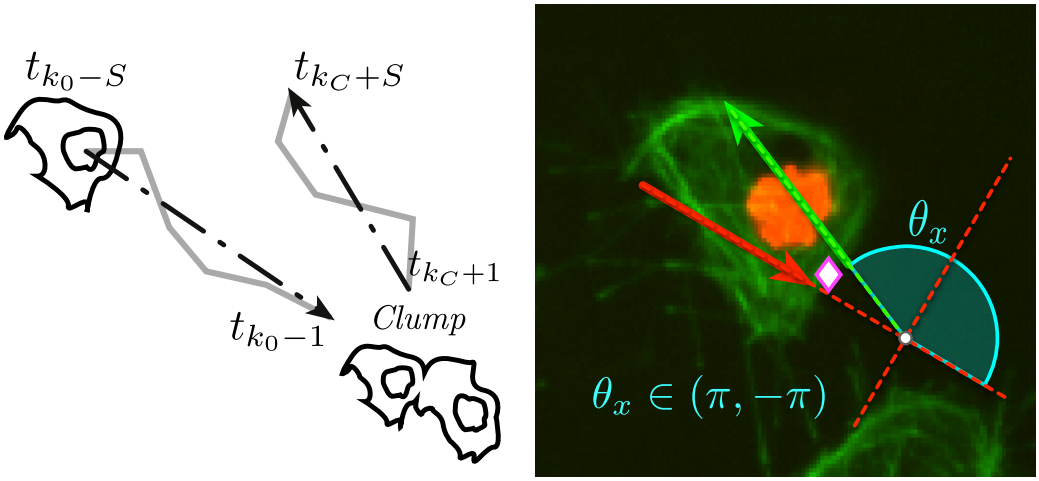


Figure 5.6 Graphical illustration of the analysis done in this work. Three principal phases are depicted. (a) Segmentation of each fluorescence channel. (b) Tracking of the red channel and identification of each clump. Each mark ($\times$) in the diagram corresponds to a different time frame. (c) Finally, the measuring of the change of direction angle (θ_x) before and after a detected clump.

Let $\theta_x \in (-\pi, \pi)$ be the angle that measures the direction change (Δ direction). Let a track with label r, given by $T_r = \{(x_t, y_t) \in \mathbb{R}^2 | t = t_1, \cdots, t_T\}$, interact with another T_q through a *clump* with code $c(r, q)$, such that the overlap between the two cells happens at time frames $t_{k_0}, t_{k_1}, \cdots, t_{k_C}$.

Measuring the change of direction before and after a clump

The algorithm developed in this work, estimates the angle formed between the direction of the cell prior to an interaction (*clump*) and the direction of the cell once the interaction is over and it does not belong to any clump. Let $\theta_x \in (-\pi, \pi)$ be the angle that measures the direction change (Δ direction). Let a track with label r, given by $T_r = \{(x_t, y_t) \in \mathbb{R}^2 | t = t_1, \cdots, t_T\}$, interact with another T_q through a *clump* with code $c(r, q)$, such that the overlap between the two cells happens at time frames $t_{k_0}, t_{k_1}, \cdots, t_{k_C}$. The determination of θ_x involves analysing the tracks T_r, T_q starting S frames before t_{k_0} and finalising at S frames after t_{k_C}. Frames $t_{k_0-S}, \cdots, t_{k_C+S}$ will be referred to as the *clump span*; likewise, the time frames where the tracks are interacting, $t_{k_0}, t_{k_1}, \cdots, t_{k_C}$, will be referred to as *time in clump*. Figure 5.8 shows a schematic of the tracks analysed and the choice of the time frames. The relationship between the frames t_{k_0}, S and T_{k_C} can be clearly observed as the moments in the *clump span* containing tracks $\mathcal{T}_r$ and $\mathcal{T}_q$. The moments are called (a) pre-clump, (b) clump and (c) post-clump.



(a) . (b) .

Figure 5.7 Diagram explaining the calculation of the angle of direction change (θ_x). (a) Shows the determination of the direction before and after the clump, while (b) displays the calculation of the angle θ_x from the previously selected lines. The red arrow represents the line generated by points at times t_{k_0-S} and t_{k_0-1}, while the green arrow shows the points at times t_{k_C+1} and t_{k_C+S}.

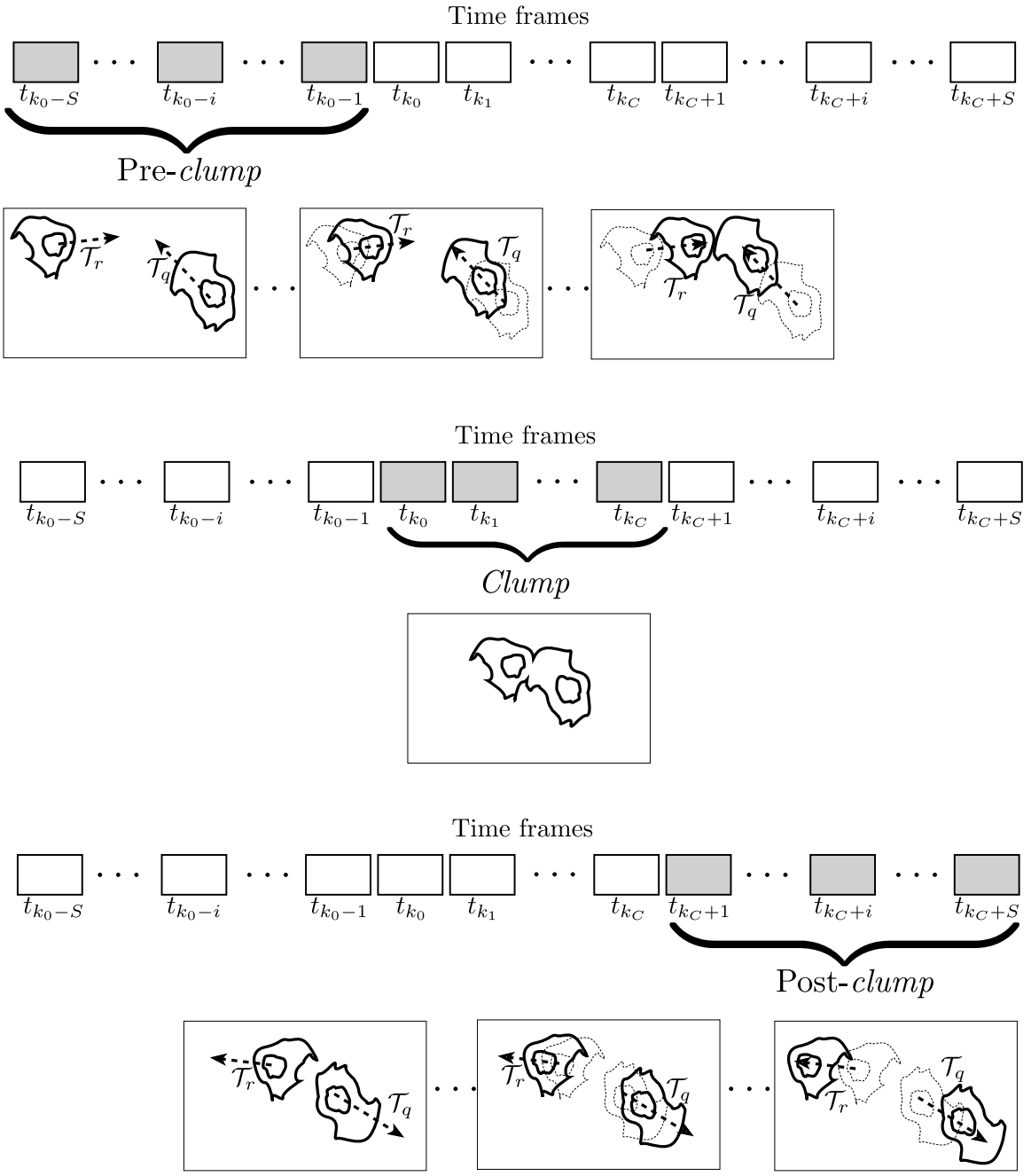


Figure 5.8 Diagram explaining the time frames chosen from tracks $\mathcal{T}_r, \mathcal{T}_q$ for the analysis of direction change. The time frames chosen for the analysis are S frames before t_{k_0} and S frames after t_{k_C}. The time frames are selected, and schematics of the cells moving are shown for each stage.

Description of experiments' results

Once the tracks involved, and the span have been manually selected, the calculation of the change of direction angle is done by selecting a vector oriented towards the clump and another one leaving it. Figure 5.7 displays the process of selecting the lines from which direction before and after the clump will be selected, and the way the angle will be measured. The estimation assesses the change of direction of each cell that exits a *clump*, relative to the orientation it has entering it. To calculate the angle, vectors must be aligned and rotated from the original positions in the image (x, y) to a new set of rotated axes (x', y'). This is performed in an intermediate step, where the incidence angle is calculated and all the points in the track are rotated to the new axes.

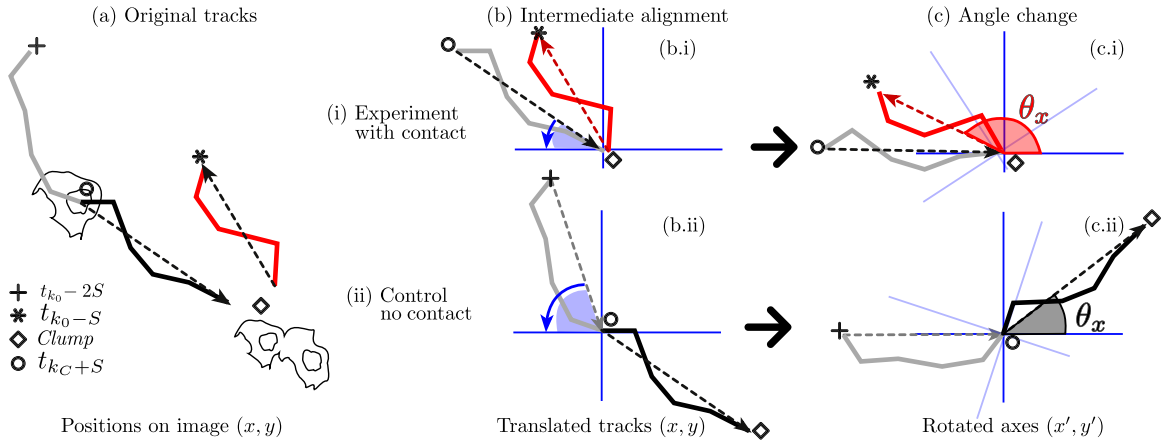


Figure 5.9 Measurement of angle change with cell-cell contact (i) and control (ii) experiments. Four markers are highlighted corresponding to specific time frames in each experiment. In order, the markers are: $(+)$ $2S$ frames before contact; $(\circ)$ S frames before contact; $(\diamond)$ starting instant of the clump; and $(*)$ S frames after the clump has finished. (a) Shows the original image with their original positions (x, y). (b) Shows the intermediate rotation of the tracks, where the tracks are rotated onto a common axis. (c) Represents the actual calculation of θ_x.

Figure 5.9 represents the calculation of the angles compared for the interactions and the control experiments. It shows how to interpret the results in Chapter 6. Notice that the new **axes** (x', y') in Figure 5.9(b) can be interpreted as a new frame of reference, containing all the positions rotated and aligned.

5.4 Summary

Two different analyses and their corresponding algorithms, were presented in this chapter; to analyse the evolution of the shape of macrophages while migrating and to understand the role that interaction of cells have on the path of them. Alongside the baseline segmentation presented in chapter 4, the two algorithms form a framework centred around PhagoSight [102], and the keyhole tracking algorithm to link the positions in the cells' nuclei from frame to frame.

The experiments and results of the methods described in this chapter are presented in Chapter 6. However, some considerations from observations and development of the algorithms can be made at this point. The **shape tracking** algorithm was originally designed as a way to solve the overlapping segmentation problem by incorporating the relationship between frames. The initial experiments, apart from the work presented in [108], were not satisfactory as the active contours would *leak*. Restrictions on the boundaries of the cells at each step could allow the method to be extended into segmenting all cells individually. Despite the limitations, the method is currently capable of track shape measurements and shape-states which could provide additional information to the tracks. In the **analysis of interactions**, observation of the nuclei moving shows small variations in the cells' trajectories, relative to the observed movement of the cell movement in the green channel. It is worth noting that the relative changes in position from the nuclei are not taken into consideration for the analysis, simplifying the analysis by considering the nuclei paths the overall movement of the entire cell. Also, the notion of cell-cell contact is determined by the presence of clumps, not by the first contact of the microtubules. This design choice in the segmentation algorithm. serves the purpose of analysing the overall cell shape.

Chapter 6

Experiments and Results

In this chapter, the principal experiments performed in the data are explained, and its corresponding results presented. The experiments presented include all the contributions by the author, spanning the different analyses made in the different datasets shown in Chapter 2.

The chapter is subdivided into three major sections related with (i) ISBI, the 2015 edition of the ISBI Cell Tracking Challenge [64], (ii) the analysis of movement of *Artemia franciscana* as shown in [53], and the foremost conclusions in such work and (iii) research in macrophages segmentation and movement analysis. The key research presented in this chapter is the investigations done on the macrophages images acquired by Stramer et al. [4], where various results were found. These include detection of foreground the clumps from single cells and the disambiguation of overlapping cells from clumps [71, 72], overviewed in chapter 4. From the algorithms described in chapter 5, the classification of single cells by shape, and a measurement of the elongation of the microtubules [72, 108] is presented; finally, the initial steps towards determining the change of direction of the cells as a result of cells interactions [109]. All contributions, provided both technical and biological insights, were found from the dataset. All sections describe the experiments referring to the algorithms and techniques specified in previous chapters. Then, the results are presented, making the distinction between results previously published and extensions made for this thesis or that were not published in due time. Finally, a brief discussion on the specific experiments is provided.

6.1 ISBI Cell Tracking Challenge

The 2015 ISBI Cell Tracking Challenge (CT-Challenge) was used as an entry point to cell segmentation and tracking problems. A fully automatic software was a requisite for submission to the CT-Challenge. Thus, the experiments presented in this section describe the amendments made to the PhagoSight software to make it fully automatic, which were described in Appendix C. Difficulties found in the data involved the automatic detection of optimal thresholds for each dataset, and in particular, an extra preprocessing step required for the dataset **C2DL-PSC**, which was acquired via phase contrast microscopy. Experiments are described and qualitative results presented, the submitted data was used, which did not include ground truth to make segmentation or tracking comparisons; however, a visual approach suffices to extract some insights from the dataset and the overall participation in the CT-Challenge.

Experiments

Three experiments are presented: (i) detecting automatic thresholds per dataset to use when running PhagoSight; (ii) segmentation and tracking results, where some example segmentation frames are presented and tracks shown; (iii) description of an additional preprocessing step applied in the **C2DL-PSC** dataset.

Finding appropriate thresholds for each dataset. Preliminary experiments showed that the automatic thresholds detected by PhagoSight, (lt_{otsu}, ht_{otsu}), were too high for most of the ISBI datasets. In normal use, PhagoSight allows for user modification of the thresholds. However, for submission to the CT-Challenge, a lineal modification led to the new parameters (LT, HT), which depended of the parameters α, β, shown in equation (C.2). To find the parameters, the sum of the evaluation codes for segmentation and tracking provided by the CT-Challenge organisation was used as an objective function (equation (C.3)) to an implementation of the Particle Swarm Optimisation algorithm [112]. Each dataset was evaluated, producing individual parameters for each.

Shading correction on phase contrast dataset. A special case had to be made for the **C2DL-PSC** dataset, in which the imaging technique used was phase contrast microscopy. As it can be observed in Figure 2.9, the image profile shows noticeable differences in the background intensities towards the edges of the figure, making thresh-

olding insufficient to segment such images. Therefore, a shading correction algorithm [84] was implemented on this particular dataset before detecting the thresholds.

Segmentation and track results. In Section 2.2, 2.4 shows all datasets in the Cell Tracking Challenge and is repeated here (figure 6.1) for convenience. In this section, a qualitative output is presented from the segmentation results and with all the datasets arranged similarly in the image, but with the tracks superimposed.

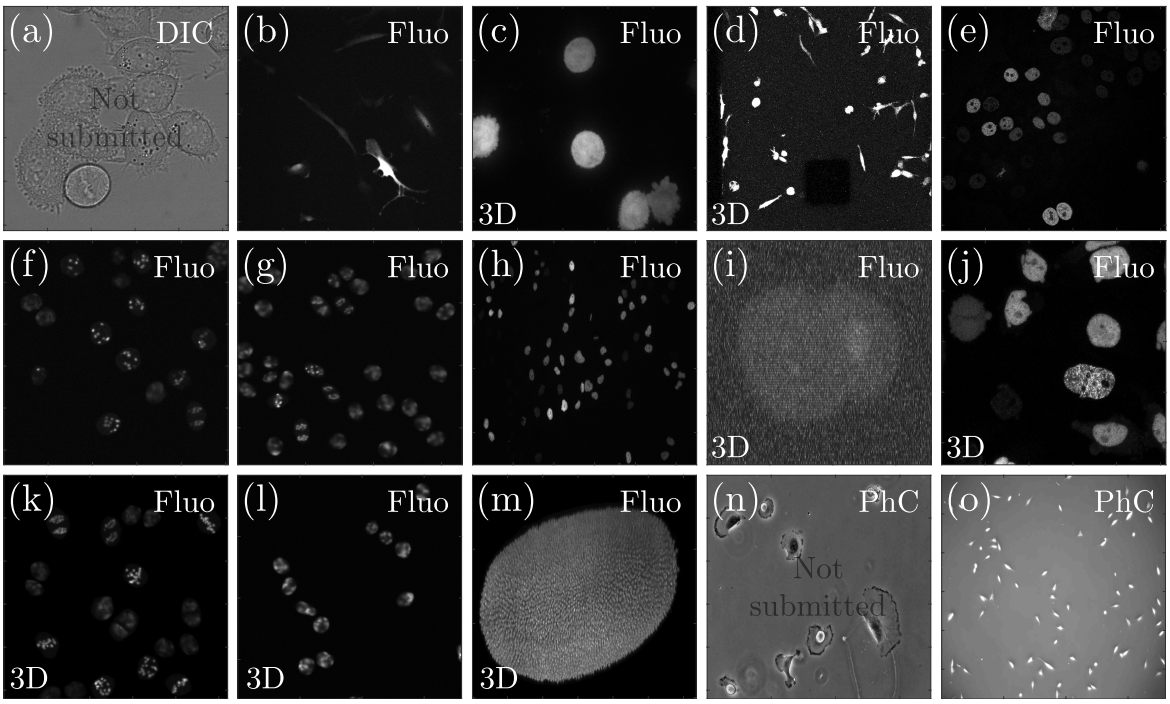


Figure 6.1 Example frames of all the ISBI datasets in the 2015 challenge (repeated Figure 2.4 for convenience). The acquisition techniques in each dataset are referred in the top right corner: DIC refers to Differential Interference Contrast, Fluo to Fluorescence microscopy and PhC refers to Phase Contrast microscopy. The top right corner shows the imaging technique. In datasets which contain 3D images display the maximum intensity projection. Dataset names are listed below, the ones submitted to the challenge appear in bold: (a) C2DH-HeLa, (b) **C2DL-MSC**, (c) **C3DH-H157**, (d) **C3DL-MDA231**, (e) **N2DH-GOWT1**, (f) **N2DH-SIM**, (g) **N2DH-SIM+**, (h) **N2DL-HeLa**, (i) **N3DH-CE**, (j) **N3DH-CHO**, (k) **N3DH-SIM**, (l) **N3DH-SIM+**, (m) **N3DL-DRO**, (n) C2DH-U373, (o) **C2DL-PSC**. Specification of the acronyms in Table 2.1.

Results

Finding appropriate thresholds for each dataset. Table 6.1 shows the comparison of the automatically produced thresholds against manually selected and the

ones detected by PhagoSight on its own. In the table, the values of four datasets are compared. The thresholds compared are: (i) thresholds defined by PhagoSight (Original PhS), (ii) threshold selected manually and (iii) the new automatic thresholds selected by equation (C.2) ((α, β)).

Table 6.1 Comparison of the ISBI Challenge results when changing the method of defining the hysteresis thresholds. For each dataset, the results of the `SEGMeasure` and `TRAMeasure` were compared depending on the way the hysteresis thresholds were defined. The first column shows the name of the dataset. The second column shows the method of defining the thresholds between **Original PhS** involving the Otsu-based algorithm, **Manual** and (α, β) for the modified thresholds described on equation (C.2). The Total column corresponds to the sum of the segmentation and tracking outputs, the highest total values are highlighted in blue to indicate the best performing method. Time is shown in seconds.

Dataset	Thresholds	`SEGMeasure`	`TRAMeasure`	Total	Time [s]
C2DL-MSC	Original PhS	0	0.019	0.019	30.845
	Manual	0.074	0.361	0.436	46.973
	(α, β)	0.250	0.613	0.863	117.520
N2DH-GOWT1	Original PhS	0.039	0.068	0.107	88.046
	Manual	0.672	0.916	1.588	400.300
	(α, β)	0.457	0.720	1.176	200.860
N2DH-SIM	Original PhS	0	0.000	0.000	32.880
	Manual	0.848	0.873	1.721	111.910
	(α, β)	0.948	0.927	1.875	92.677
N2DL-HeLa	Original PhS	0.150	0.217	0.367	85.355
	Manual	0.518	0.665	1.183	189.330
	(α, β)	0.518	0.658	1.176	191.112

Shading correction on phase contrast dataset. Figure 6.3 shows the result of implementing the shading correction algorithm in a frame. In this chapter, a comparison between adaptive thresholds seen in Chapter 3, by Pappas [82] and Bradley and Roth [83], and the shading correction algorithm are presented. This comparison was not explored in preparation for the CT-Challenge. Although the outcome of the adaptive threshold by Pappas [82] shows promising results, running time of the algorithm shows a clear disadvantage. The time consumed processing a single frame were **0.6739** (s) for shading correction [84], 6.5480 (s) for Pappas [82] and 5.2207 (s) for Bradley and Roth [83].

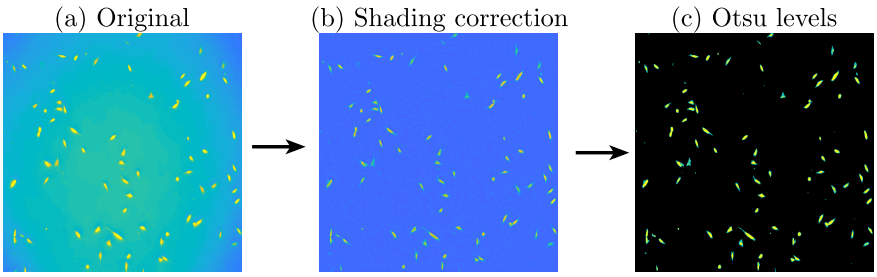


Figure 6.2 Preprocessing by shading correction applied to phase contrast dataset. (a) Original frame, the colour map chosen to highlight the shading in the imaging. (b) Output of the shading correction [84]. (c) Shows the classes of intensities after classifying them with the Otsu levels.

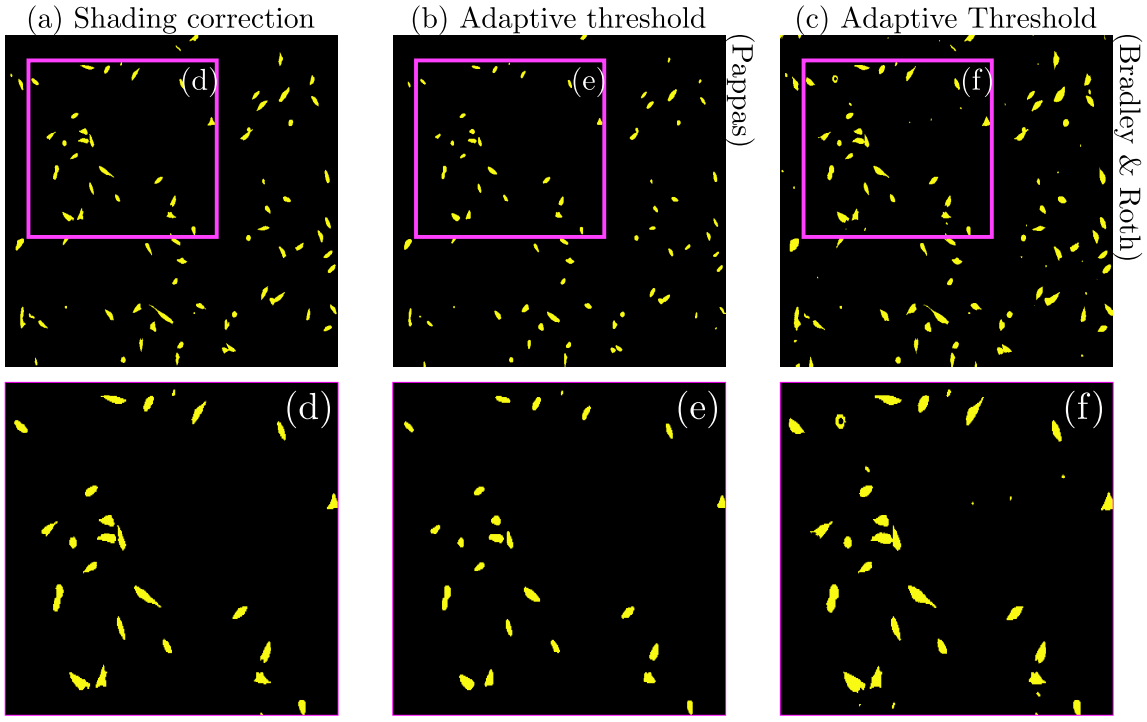


Figure 6.3 Comparison between shading correction and adaptive thresholding. (a) Otsu classes after applying the shading correction algorithm. Adaptive thresholding techniques were implemented: (b) Pappas [82] and (c) Bradley and Roth [83].

Segmentation and tracking results. Parallel to Figure 2.4, the presentation of all datasets and the results are shown for segmentation (Figure 6.4) and tracking (Figure 6.5). In both figures, the images shown correspond to a specific time frame ($t = 10$) only where segmentation and tracking outputs were produced. Particular to the segmentation results, 3D dataset results are presented by showing the maximum

intensity projection of the corresponding 3D frame. The jet colour map only represents the arbitrary cell label in the image.

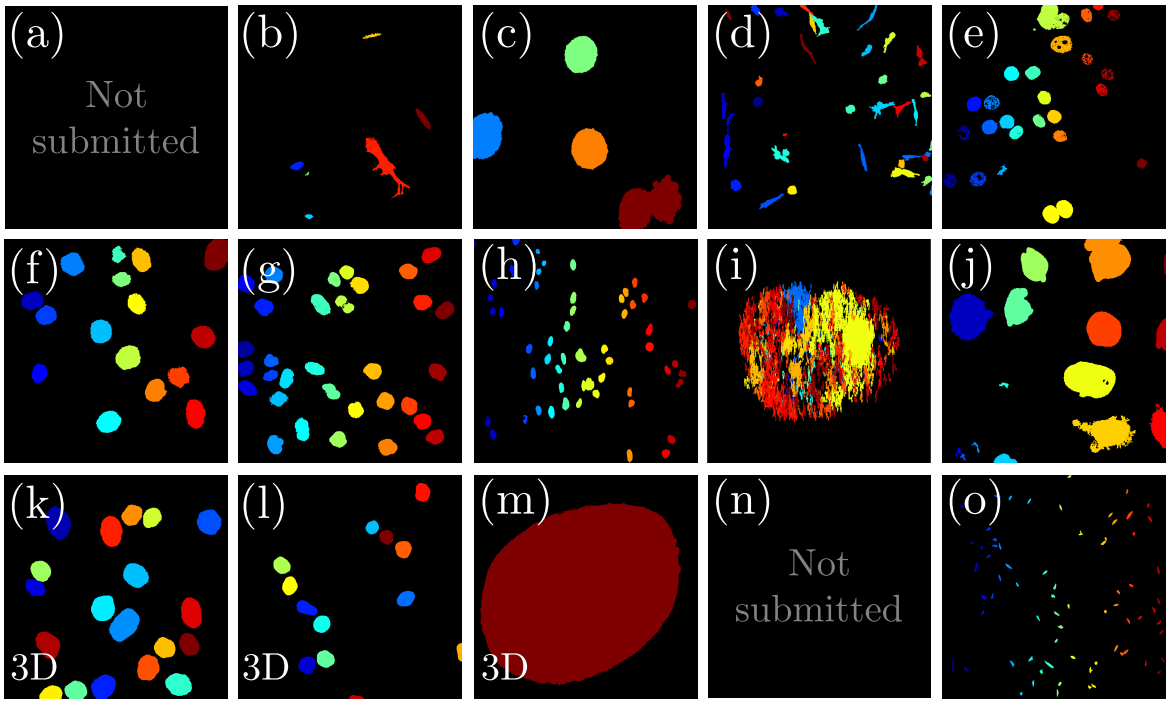


Figure 6.4 Segmentation results submitted to the ISBI Cell Tracking Challenge. Frames displayed in black are the ones where no submission was made. In datasets which contain 3D images display the maximum intensity projection. Dataset names are listed below, the ones submitted to the challenge appear in bold: (a) C2DH-HeLa, (b) **C2DL-MSC**, (c) **C3DH-H157**, (d) **C3DL-MDA231**, (e) **N2DH-GOWT1**, (f) **N2DH-SIM**, (g) **N2DH-SIM+**, (h) **N2DL-HeLa**, (i) **N3DH-CE**, (j) **N3DH-CHO**, (k) **N3DH-SIM**, (l) **N3DH-SIM+**, (m) **N3DL-DRO**, (n) C2DH-U373, (o) **C2DL-PSC**.

Regarding the tracking results, positions where a cell can be observed, but no track is found, it due to a failure in the segmentation, an example of this is dataset **N3DL-DRO**(m). Also in Figure 6.5, some results are worth noticing and should be pointed out. First of all, datasets (a) C2DH-HeLa and (n) C2DH-U373 were not submitted and do not appear in the results. Secondly, some cells in some datasets can be observed although in varying intensities, thus missing or faulty segmentation was found in such frame, and in turn, some tracks appear to be missing or abnormal. This caused different outcomes in the results. Some interesting points are highlighted in yellow arrows.

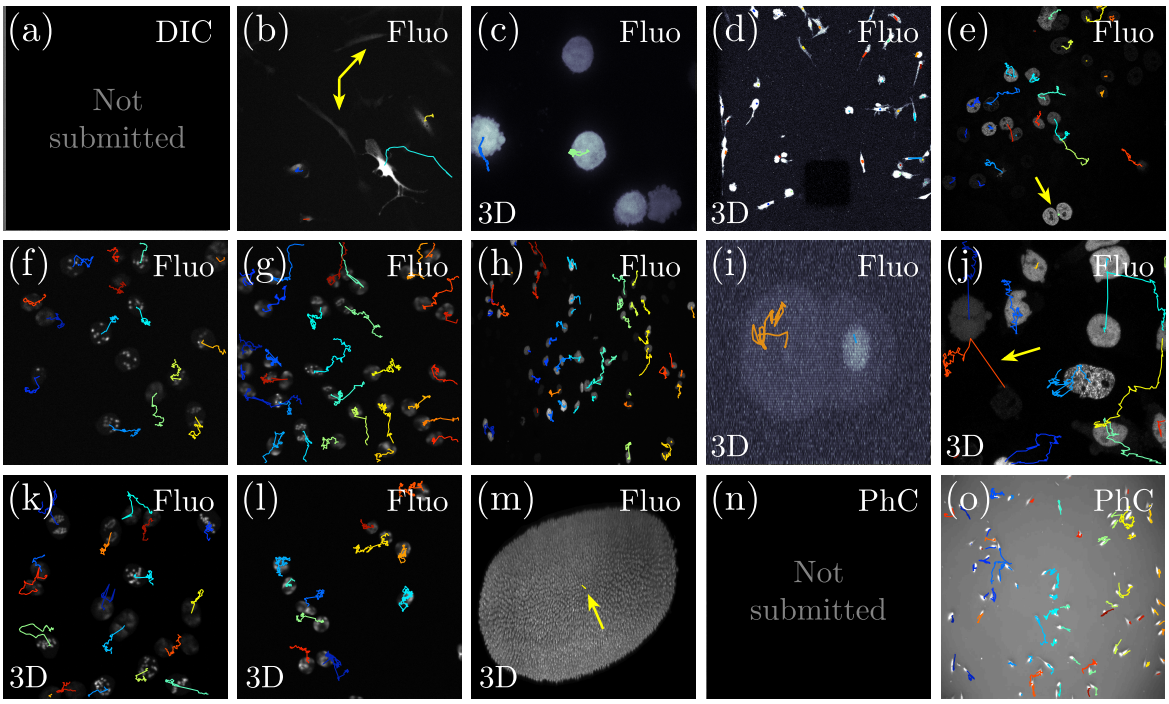


Figure 6.5 Tracking results submitted to the ISBI Cell Tracking Challenge. Frames displayed in black are the ones where no submission was made. In datasets which contain 3D images display the maximum intensity projection. Dataset names are listed below, the ones submitted to the challenge appear in bold: (a) C2DH-HeLa, (b) **C2DL-MSC**, (c) **C3DH-H157**, (d) **C3DL-MDA231**, (e) **N2DH-GOWT1**, (f) **N2DH-SIM**, (g) **N2DH-SIM+**, (h) **N2DL-HeLa**, (i) **N3DH-CE**, (j) **N3DH-CHO**, (k) **N3DH-SIM**, (l) **N3DH-SIM+**, (m) **N3DL-DRO**, (n) C2DH-U373, (o) **C2DL-PSC**. Arrows show points of interest, full explanation in text.

In dataset (b) **C2DL-MSC**, the variation of intensities is sizeable, causing cells to be missed from the analysis. Then, in dataset (e) **N2DH-GOWT1** some cells are also missing due to their intensity, but also two cells are pointed at with an arrow, because the algorithm was not able to segment those cells individually, therefore only one track is visible. Finally, in dataset (j) **N3DH-CHO** some large *jumps* form the tracks appear, and could be instances where the segmentation was not robust enough and caused the tracking algorithm to link the cells in time incorrectly. Another special case observed concerns dataset (m) **N3DL-DRO**, where the algorithm only detected a single large cell. which produced a single track. Finally, notice the shading in dataset (o) **C2DL-PSC**, the following section explains the experiments made and the extra preprocessing stage implemented to overcome it.

6.2 *Artemia franciscana* dataset

In [53], a framework was developed to investigate the relationship between the swimming patterns of the *Artemia franciscana* and the toxicity of the environment. The algorithm involves the pipeline of semi-automatic image processing steps for the movement analysis. As mentioned in Chapter 3, Figure 3.38, the mean frame was subtracted from all time frames, and negative values were rectified. This produced an appearance similar to fluorescence, where *artemia* look much brighter than the background. PhagoSight was used to segment and track the preprocessed frames.

Experiments

Fixing broken tracks. The preprocessing method caused some artefacts and noise to be generated in the images, which caused some of the *artemia* to disappear and thus producing broken tracks. Tracks that were separated by one or two frames within a small distance were assumed to correspond to the same moving object; and thus were joined as a single track. Furthermore, tracks incorrectly produced due to the detection of an artefact in the preprocessed images were manually discarded.

Calculating turns of the *artemia*. Counting individual 360° turns was of special interest, as it can help distinguish patterns or movement. The orientation angle is calculated per object at each time frame, the cumulative angle per object was recorded. Every time the cumulative angle crosses a threshold of $\pm 2\pi$, a turn is recorded.

Results

Fixing broken tracks. The results for the tracks' before and after applying the postprocessing routine are presented in Figure 6.6. The result involves 5 objects tracked throughout the 1800 frames, where several 360° turns can be observed. Figure 6.7(a) displays a single track throughout the first half of the data (900 frames) where the turns can be clearly observed. Figure 6.7(b) shows a detail of all tracks in the first 50 time frames with an superimposed frame.

Calculating turns per object. One track was selected and the distance, calculating the angle per frame, cumulative angle and distance form the origin. Figure 6.8 displays such results. Notice Figure 6.8(a), how the cumulative angle around the 1600th frame

increases consistently and turns various times in succession. angle per frame, shown in figure 6.8(b), is used to calculate when full turns of 360° occur; the relative vertical position corresponds to clockwise or counter clockwise turns.

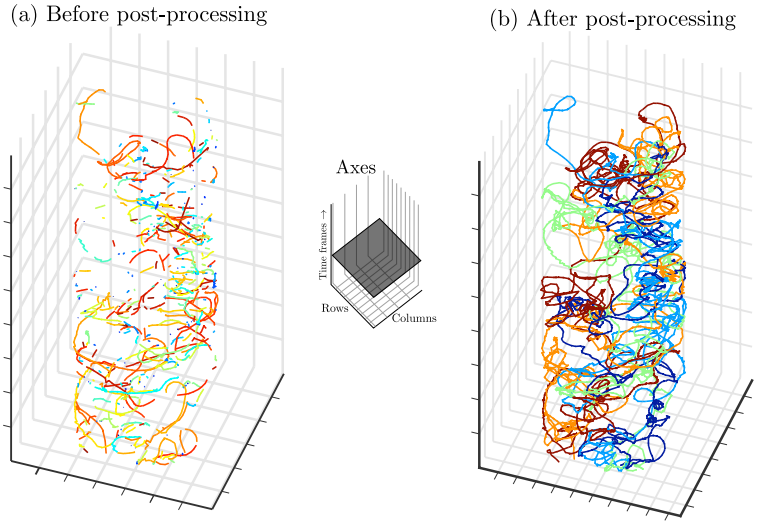


Figure 6.6 Postprocessing of the tracks resulting from inconsistencies in the segmentation. (a) Original output from PhagoSight where multiple short tracks, corresponding to artefacts detected by the segmentation, can be observed. Longer tracks get cut short, due to an object not being detected by the segmentation. (b) shows the output of the postprocessing of the tracks.

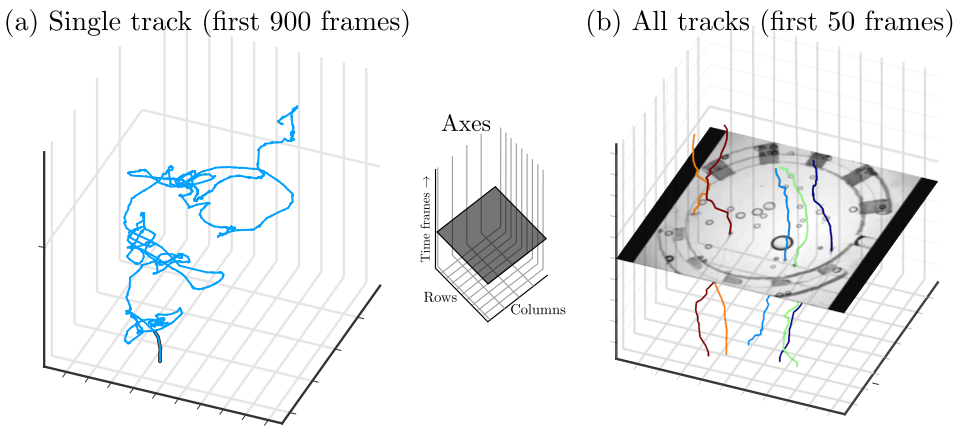


Figure 6.7 Detail of the tracks in the *Artemia franciscana* dataset. (a) Shows the first half (900 frames) of a single track. The time is on the vertical axis, moving from bottom to top. The movement appears to be a spiral where each turn occupies a larger area. (b) Represents the first 50 frames of all tracks. Frame 25 is superimposed in the image for visualisation purposes.

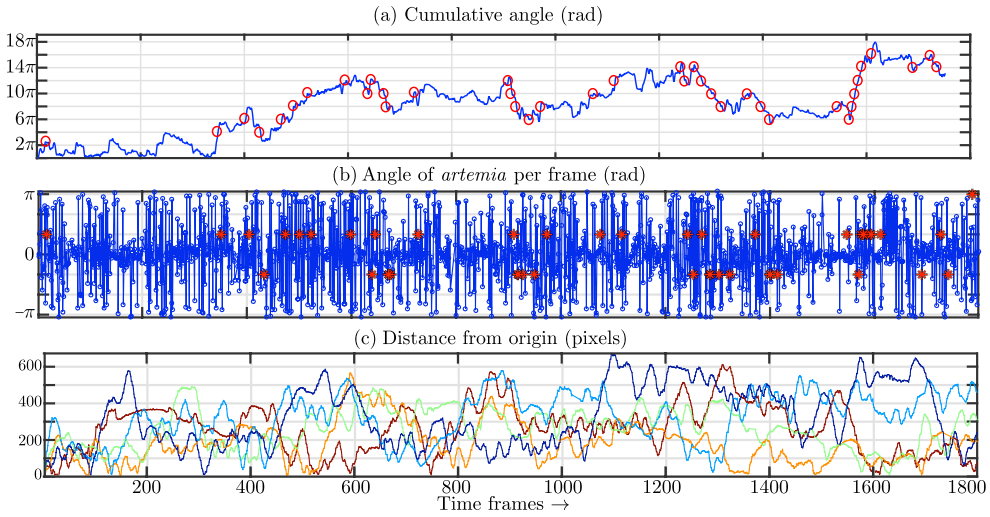


Figure 6.8 Illustration of the visualisation of measurements. (a) Cumulative angle detecting multiples of 2π turns (red $\circ$). (b) Angle at every time frame. The positions at which a complete 360-degree turn occurs are marked (red $\star$); the relative vertical position corresponds to clockwise or counter clockwise turns. (c) distance covered from the origin for all the tracks.

6.3 Detection of peaks with Anglegram

In Chapter 4, Table 4.2 shows the accuracy of the method to detect corners in synthetic shapes, the result was reported in [108]. In this section, the result will be extended to include a comparison with the Harris' corner detector [103]. It is worth noting that the Harris corner detector ranks all corners based how sharp or *pointy* the junctions are detected, assigning them a number between 0 and 1, called metric.

One thousand different synthetic shapes were generated. The peaks in each shape were detected with the *anglegram*, and the number of peaks stored, as the number of peaks defines the type of shape: zero for circle, 1 for drop, 2 for bidrop and 3 for tridrop. In order to have a fair comparison with the Harris algorithm, an additional procedure was implemented to detect the specific number of junctions, depending on the basic shape. The strongest five junctions detected by Harris' method with a metric higher than a certain threshold were selected providing the number of peaks. From visual inspection of the metrics of the strongest corners, four thresholds (0.06, 0.07, 0.08 and 0.09) were applied and the number of junctions whose metric exceeded the threshold were considered as detections.

Table 6.2 Peak detection using the Harris algorithm. The metric values of the five strongest junctions were tested against different threshold values. Each column corresponds to the number of peaks detected, and the rows correspond to the basic shape the detection represents. The highest values per shape were highlighted and presented in Table 6.3. Results were obtained from a synthetic dataset from 1000 shapes.

	Harris metric > 0.06					Harris metric > 0.07				
Shapes /	Peaks detected (%)					Peaks detected (%)				
peaks	None	1	2	3	≥ 4	None	1	2	3	≥ 4
Circle	87.88	9.85	2.27	0	0	91.67	8.33	0	0	0
Drop	5.65	77.82	13.31	2.42	0.81	8.47	**81.05**	8.06	1.61	0.81
Bidrop	3.44	33.59	**56.87**	5.73	0.38	6.87	36.26	53.82	3.05	0
Tridrop	0	2.21	26.55	**61.06**	10.18	0	3.10	33.63	58.85	4.42
	Harris metric > 0.08					Harris metric > 0.09				
Shapes /	Peaks detected (%)					Peaks detected (%)				
peaks	None	1	2	3	≥ 4	None	1	2	3	≥ 4
Circle	97.35	2.65	0	0	0	**98.48**	1.52	0	0	0
Drop	19.35	75.00	4.03	1.61	0	21.77	74.19	2.42	1.61	0
Bidrop	18.32	47.33	33.59	0.76	0	20.61	47.71	31.30	0.38	0
Tridrop	0.44	15.04	46.02	37.17	1.33	1.77	14.60	46.90	35.84	0.88

Overall, the *anglegram* method had an accuracy of 92.9%, while the algorithm by Harris' accuracy ranged from 59.9% on the 0.09 threshold to an overall accuracy of 71.347%, when using the 0.07 threshold. For the anglegram method, tests on the synthetic data were in the range of 87.5-99.2% correct detection, as shown in Table 4.2. The method achieved the highest accuracy in the **bidrop** shape, and the lowest in the **drop** shape. The **circle** shape detection improved considerably since the amendment made to the method reported in [108].

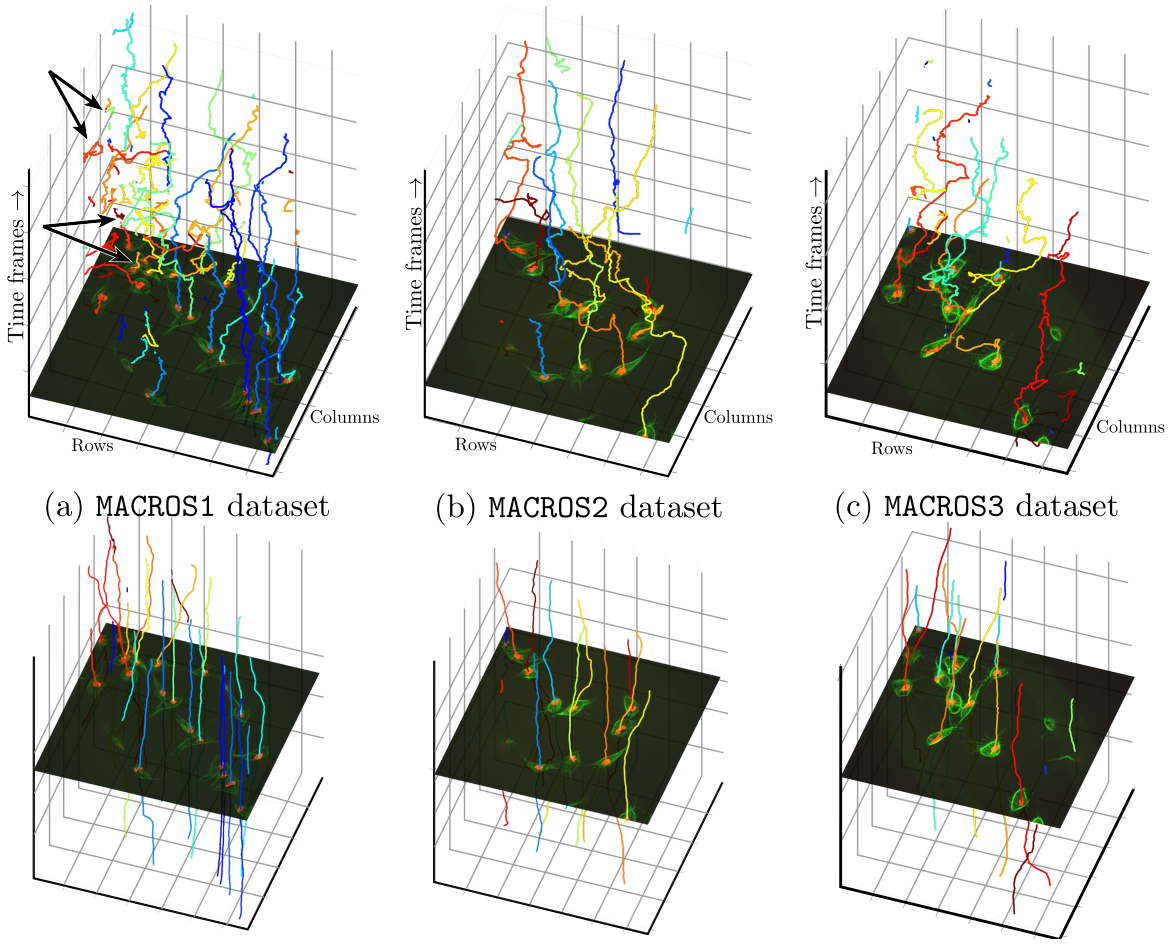
Table 6.3 Comparison of the number of peaks detected by the *anglegram* and the Harris corner detector. The best column per experiment in table 6.2 to simulate a best case scenario for comparison. The accuracy of each method is highlighted. Results were obtained from a synthetic dataset from 1000 shapes.

Shapes /	Number of peaks detected (%)					Number of peaks detected (%)				
peaks	None	1	2	3	≥ 4	None	1	2	3	≥ 4
Circle	**99.62**	0	0	0.38	0	**98.48**	8.33	2.27	0	0
Drop	4.44	**87.5**	6.05	2.02	0	21.77	**81.05**	13.31	2.42	0
Bidrop	0	0.38	**95.8**	3.82	0	20.61	36.26	**56.87**	5.73	0
Tridrop	1.33	0	2.65	**88.94**	0	1.77	3.1	26.55	**61.06**	0.88
						$> .09$	$> .07$	$> .06$	$> .06$	$> .09$

Table 6.3 displays an inviable best-case-scenario for the junctions detected by Harris, compared to the results from the peak detection methodology using the anglegram. Results from the anglegram method are higher in all categories. Furthermore, the anglegram method ignores corner junctions, where the prevailing angle is obtuse whereas the Harris algorithm would not make such a distinction, and in more complex shapes more junctions would be detected.

6.4 Fluorescently labelled macrophages analysis by the `macrosight` framework

An overview at the three macrophages datasets is shown in figure 6.9, where all the detected tracks from the red channel are presented for all the frames in each dataset. The first 100 frames per dataset are presented in the bottom row. In general, short tracks hint to issues resulting from the image acquisition, segmentation or tracking. In the case of the macrophages data, cells close to the upper right and lower left corners are prone to errors because the cells are not visible in those positions.



(a) MACROS1 dataset (b) MACROS2 dataset (c) MACROS3 dataset

Figure 6.9 Complete tracks (**top**) obtained from red channel in all three datasets and detail of the first 100 tracks (**bottom**) per dataset. The datasets are organised in columns: (a) MACROS1, (b) MACROS2 and (c) MACROS3. Track colours in all cases represent the relative speed of the objects, following jet colour map with blue - slow, red - fast. (a) MACROS1 displays the higher amount of inconsistent tracks (black arrows).

The results of the algorithms described Chapter 5 are presented in this section. The experiments and results of the shape tracking and analysis, as described in Section 5.2, will be described in Section 6.4. The analysis of interactions, resulting from the algorithm described in Section 5.3, will be described in Section 6.4.

Shape evolution and tracking

The experiments correspond to the shape evolution algorithm with some of the single cells. Three experiments were performed in the MACROS1 dataset in which a cell is *followed* alongside a track, analysing the shape as described in Section 5.2. Figure 6.10

shows an overview of the three experiments presented in this section. Three tracks were followed in a span of 50 frames or less.

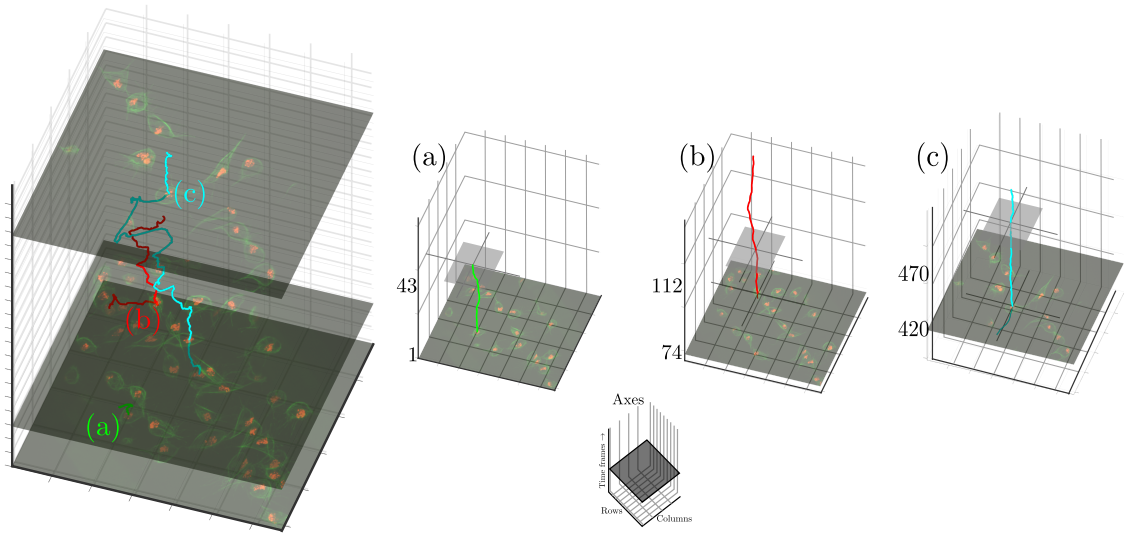


Figure 6.10 Overview of the three experiments presented in this section regarding the shape tracking analysis. The horizontal axes represent the rows and columns. The vertical axis represents time, and each section represents 25 frames. **Left column** shows three complete tracks throughout the 536 time frames. The analysis has been done in segments of the tracks where the cells were not in clumps. On the **right**, a detail of each cell track is shown.

Figure 6.11 displays the shape evolution of the three track-fragments presented before, containing 43, 38 and 50 frames, respectively (a,b,c). Each experiment show three rows marked (i,ii,iii). Per experiment, eight frames are shown with the moved boundary displayed in the dotted line (cyan- -) and the evolved shape in the solid line (magenta-), **top row** (i). For visual assessment, the peaks detected by the anglegram at each point are displayed as asterisks (yellow $*$). The **middle row** (ii) shows a comparison between the orientation (blue $*-$) and the aspect ratio (orange $+-$). The values for the top frames are highlighted in blue($\circ$) and black ($\diamond$). The **bottom row** (iii) shows the minimum intensity projection of top's respective *anglegrams* displaying the values of the angles measured per corner (red $\diamond$).

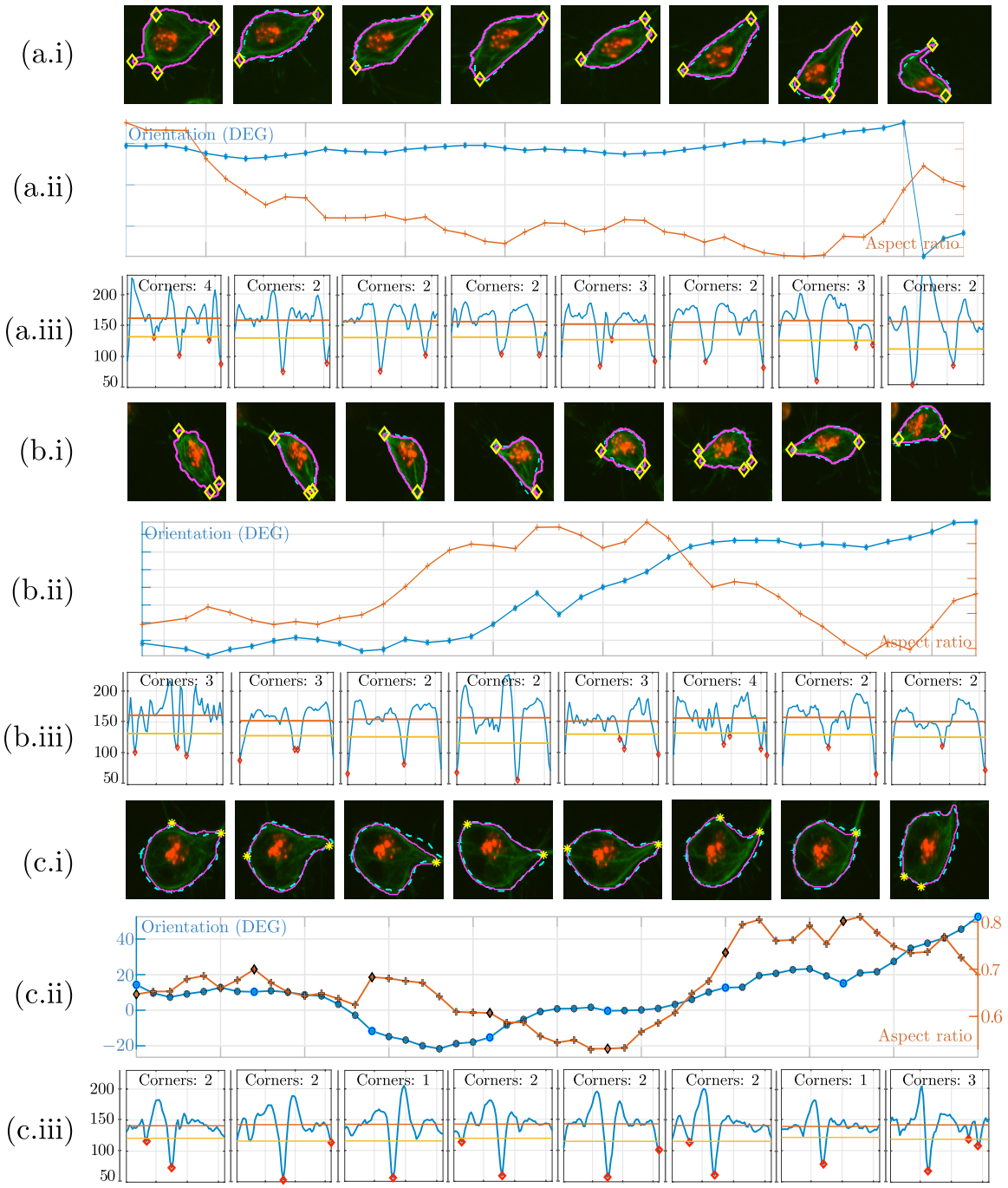


Figure 6.11 Evolution of shapes throughout multiple frames. The indexes (a,b,c) corresponding to the indexes in Figure 6.10. Full explanation of the rows in text. (i) Represents eight frames in the experiment with the boundary superimposed (current - magenta, previous - cyan), as well as the peaks detected (yellow ⋄). (ii) Orientation (DEG) and aspect ratio computed throughout the shapes in the experiment. (iii) Shows the *mIP* of the anglegram, and the detected minima, which correspond to positions in the boundary showing a peak. Notice the peaks detected in (a.i) where the algorithm successfully detected two peaks in the final frame, despite being a complex shape. The first and fifth frames show extra peaks detected. Frames in (b.i) show a similar problem, with two very close peaks are detected.

Analysis of direction changes

All three datasets were segmented and tracked, however, only two datasets (`MACROS2` and `MACROS3`) were considered for the change of direction analysis due to inconsistent tracks being present in `MACROS1`, where a single cell would produce more than one tracks. The reader is referred to Figure 6.9(a) where several short and inconsistent tracks can be observed. The tracks' information was searched to find cases of clumps that fulfil the following criteria:

1. **Only two cells interacting.** There are cases where more than one cells integrate a clump, e.g Figure 5.2. These cases were excluded from the analysis as it is not clear whether the interaction of more than two cells would be different from the interaction between a pair.

2. **In and out cases.** The cases selected only involved cells with a well-defined *clump span*, in which the cell would enter the clump and exit it without disappearing or interacting with other clumps.

3. **Immediate reaction.** A small value for $S = 5$ was chosen to define the *clump span*, as the interest of this work is to study the immediate reaction of a cell after interacting within a clump. In this work, the *clump length* was not taken into consideration.

4. **Both cells in clump.** Cases where both tracks in the clump had a well-defined *clump span* were preferred as they would allow an analysis per *clump*.

Once the tracks were selected, the θ_x angles were calculated for each case. Additionally, for each track selected, the $2S$ time frames leading up to the *clump* were selected to compare the change of direction with a cell that has not interacted with a clump. These tracks will be referred to as **control tracks** or **single cell movement**. The experiments were chosen in a semi-automatic way, using the information from the table in Figure 5.2 to generate candidates of experiments and manually logging the starting and ending points $t_{k_0} - 2S, t_{k_0} - S, t_{k_0}$ and $t_{k_C} + S$ per experiment. It is important to notice that the experiments were chosen without considering the *time in clump*. As highlighted in Section 5.3, such parameter could be influential to the results, as the the window where the interaction is observed occurs within a few minutes, which would be translated to a cell belonging a small number of time frames in clump.

The original results reported in [113] are presented verbatim, then extra conclusions drawn from them are described and presented, these include dataset `MACROS3` and is shown in figure 6.12(c) statistical analysis presented next.

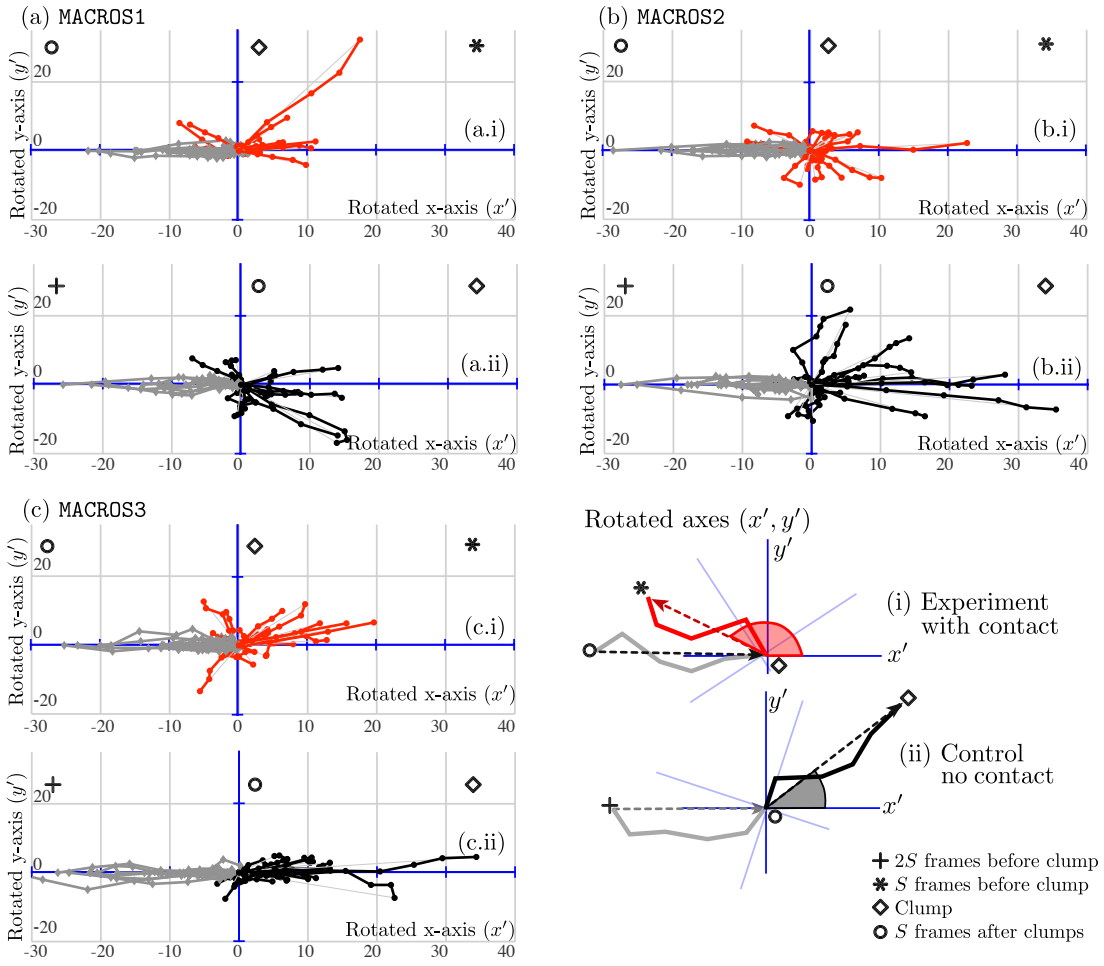


Figure 6.12 Comparison of the changes in track directions with or without clump interaction within the all datasets, (a) $N = 15$, (b) $N = 17$ and (c) $N = 20$. (c) Correspond to the results reported in [109]. The sold line represents a cell's trajectory. Each line can be read from the utmost left point and continuing along the line. For all datasets, (a) `MACROS1`, (b) `MACROS2` and (c) `MACROS3`, (i) illustrates cells entering or exiting a clump, where the origin ($\diamond$) represents the clump formation. Grey lines represent $S = 5$ time frame points of each cell's track before entering a clump. Red lines represent 5 time frame points of each cell's track after exiting a clump. (ii) illustrates the movement of cells before entering a clump, where the origin ($\circ$) represents a chosen arbitrary point (time frame 6).

All datasets were segmented and tracked. Tracks were selected based on the criteria described before. In total, fifty two cases were found with $N = 15, 17, 20$ respectively for each dataset. To represent the tracks and changes in directions for all datasets and compare them to the control tracks, Figure 6.12 is presented to qualitatively show the hypothesis depicted in Section 5.3. The figure contains a key taken from the explanatory Figure 5.9.

Several differences can be observed in the tracks containing cell-cell contact, varying depending on the dataset. To assess the changes of angles of all experiments collected, the mean and median of the angle changes were compared from experiments to control. Referring to Figure 6.12, Table 6.4 summarises the angle comparisons made with the mean and standard deviation in each case. The Wilcoxon Signed Rank test [107] was implemented in all cases, to compare the median of the measurements and a normal T-Test was implemented to compare the means.

Table 6.4 Angle change (θ_x) comparison per dataset. The mean and standard deviation angle change were calculated, and the results of the statistical tests comparing both contact and control experiments are shown. In moments where the null hypothesis could not be rejected are highlighted in red.

DATASET	Cell-cell contact mean (std)	Control mean (std)	WILLCOXON p-value	T-TEST p-value
`MACROS1`	53.79 (64.25)	-4.34 (74.18)	0.08	0.03
`MACROS2`	0.61 (77.31)	15.48 (78.10)	> 0.05	> 0.05
`MACROS3`	37.40 (77.65)	-15.59 (46.68)	0.02	0.01
ALL	30.10 (75.70)	-2.19 (66.42)	0.03	0.02

A more thorough exploration of the cases and the tracks was implemented, considering the time in clump (TC) parameter. First, consider the average TC per the datasets, where `MACROS1` had an average of $TC = 3.6 \pm 3.18$, `MACROS2` had an average of $TC = 19.65 \pm 24.96$ and `MACROS3` had an average of $TC = 12.30 \pm 14.25$. Visually, TC can be observed in Figure 6.13, which displays three examples of cells interacting through a given *span* and the orientation lines before (red) and after (green) the *clump*. Clumps shown contain the codes `2001, 3002` and `22001`, which have very different values of TC, as can be observed from the yellow lines.

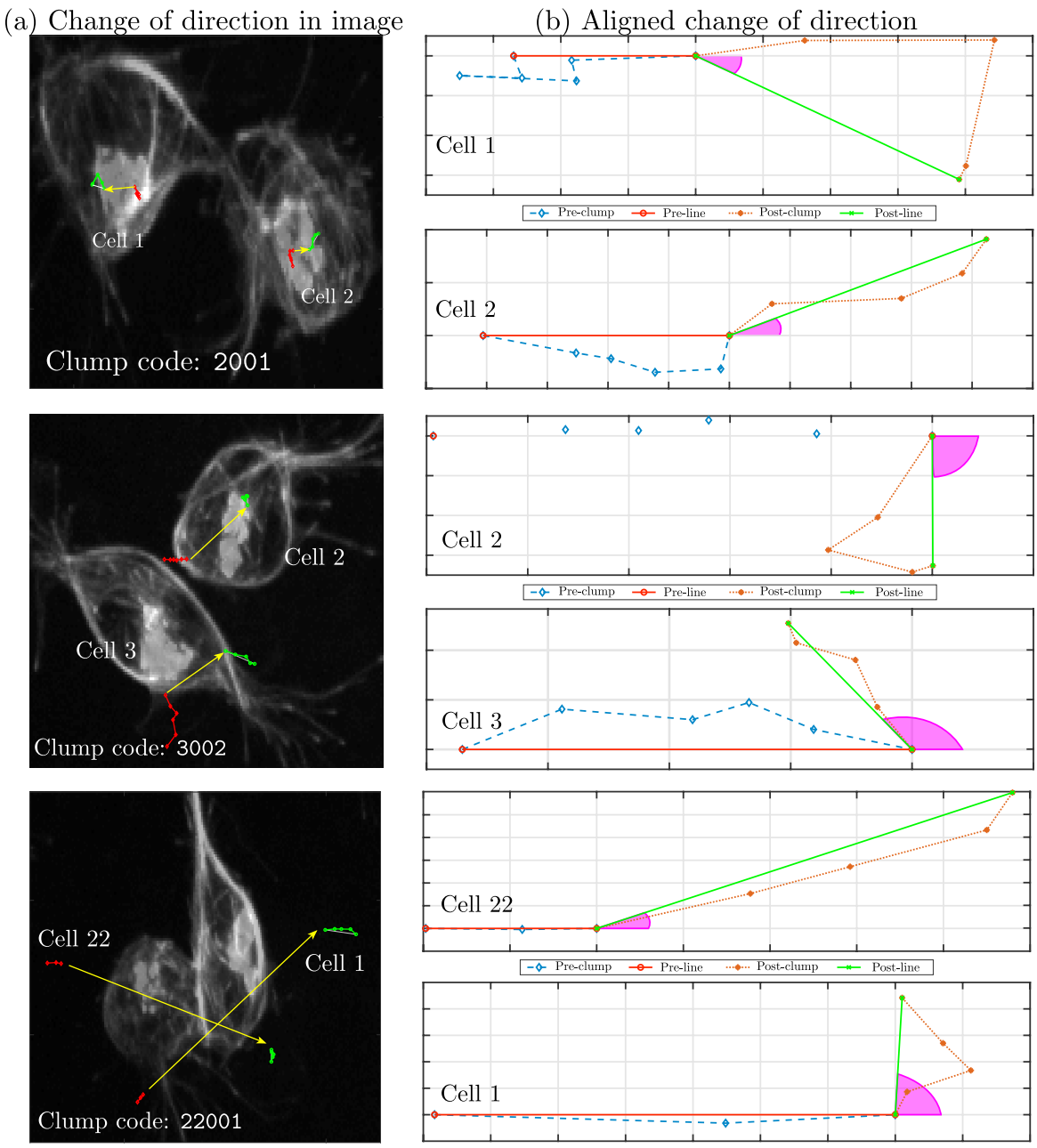


Figure 6.13 Examples of change of direction before and after a clump. Column (a) shows three examples of cells interacting in three different clumps: `2001`, `3002` and `22001`. Red line ($*-$) shows the orientation of movement before the clump, and a green line ($\diamond-$) represents the positions of movement after. A yellow arrow was superimposed on the image to show the trajectory of the cell inside the clump. (b) Simplified view of the cells' changes in orientation. The cells' path before the clump is represented in blue ($-\diamond-$). The path of the cell after the clump is shown in orange ($:*$). The angle arc of orientation is shown in magenta. Notice that the movement of the two cells involved in clump `2001` is a lot smaller compared to the other cells. For a detailed movement analysis of the cells, figures 6.16, 6.17 and 6.18.

Figures 6.14 and 6.15 explore the differences between change in angle and time in clump (TC). From figure 6.14(a), it can be observed that TC range for datasets `MACROS1` and `MACROS3` is much smaller compared to the range of the TC for dataset `MACROS2`. The scatter plot 6.14(c) shows a distinct gap in the range $20 < TC < 40$, and most of the cases in the range $0 \leq TC < 10$.

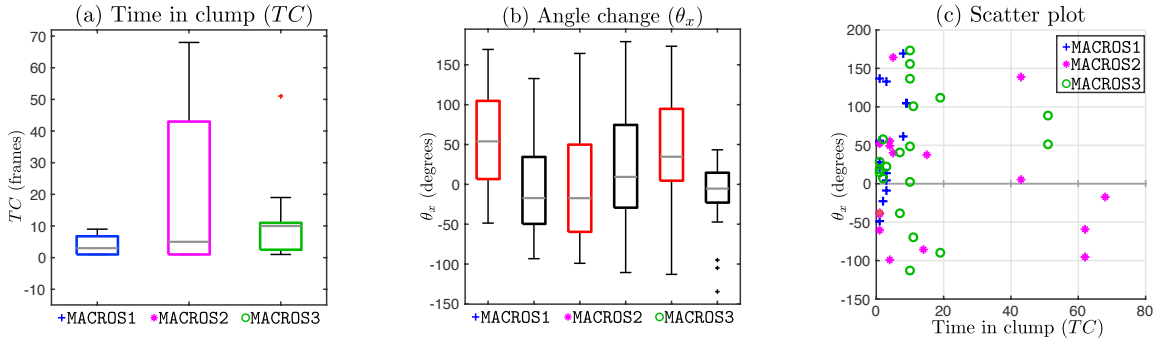


Figure 6.14 Comparison of the time in clump and the angle of variation. (a) Boxplot diagram comparing time in clump (TC) per dataset. (b) Boxplot comparing the value of the measured angle change from the contact (red) and the control (black) experiments. (c) Scatter plot of time in clump vs. angle change in contact experiments per dataset.

In Figure 6.15, a comparison was made between the angle change depending on TC. Each row represents the comparison of θ_x between cases with cell-cell contact and control cases, when selecting only cases with $TC \leq a$ where $a = 2, 6$ and 10. Although most cases are broadly similar, cases with $TC \leq 2$ and $TC \leq 6$ show a more distinctive difference between control and cell-cell contact cases, with control cases (black box plots) showing a greater range in almost all cases. A final experiment was run, in which the value of the angle change θ_x was compared between the control and interaction experiments presenting a $TC < 10$. The cases satisfying $TC < 10$ in all datasets were combined, giving a total of 33, where the mean for interaction cases was $31.65° \pm 64.05$ and the control cases was $-8.87° \pm 63.38$. The Wilcoxon Signed Rank test produced a p-value of 0.03, providing statistical significance to the difference.

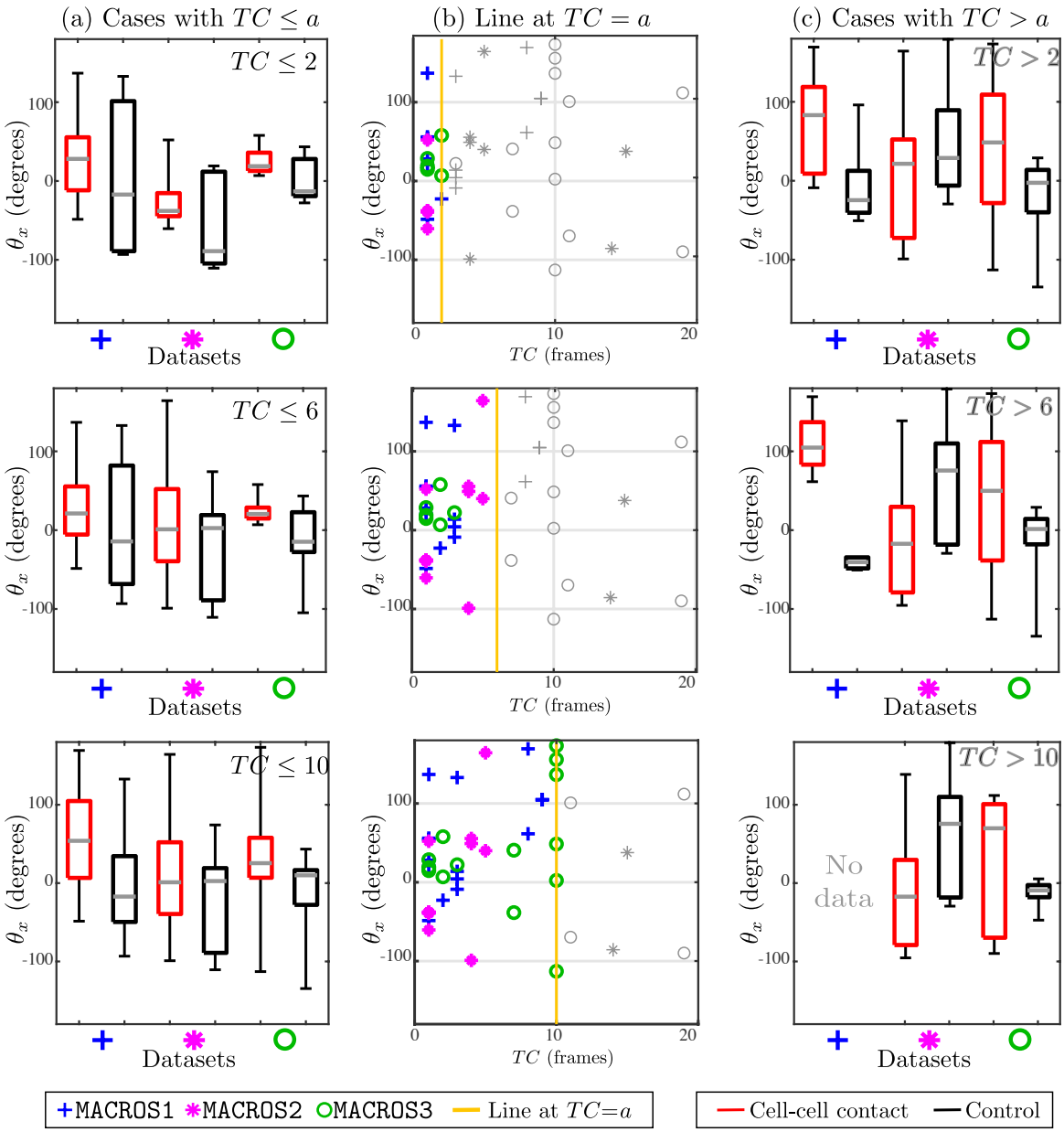


Figure 6.15 Comparison of angle change, depending on the time spent in clumps. In both datasets, boxplots of the angle change were generated for experiments with cell-cell contact (red) and control movement (black). The figure should be read by rows, where each one corresponds to a specific number of frames the cells remained in a clump (Time in clump: $TC = a, a = 2, 6, 10$). The left column shows boxplots of the experiments that fulfil the criterion $TC \leq a$ and the right column show the experiments where $TC > a$. The middle column shows scatter plots showing all cases from both datasets.

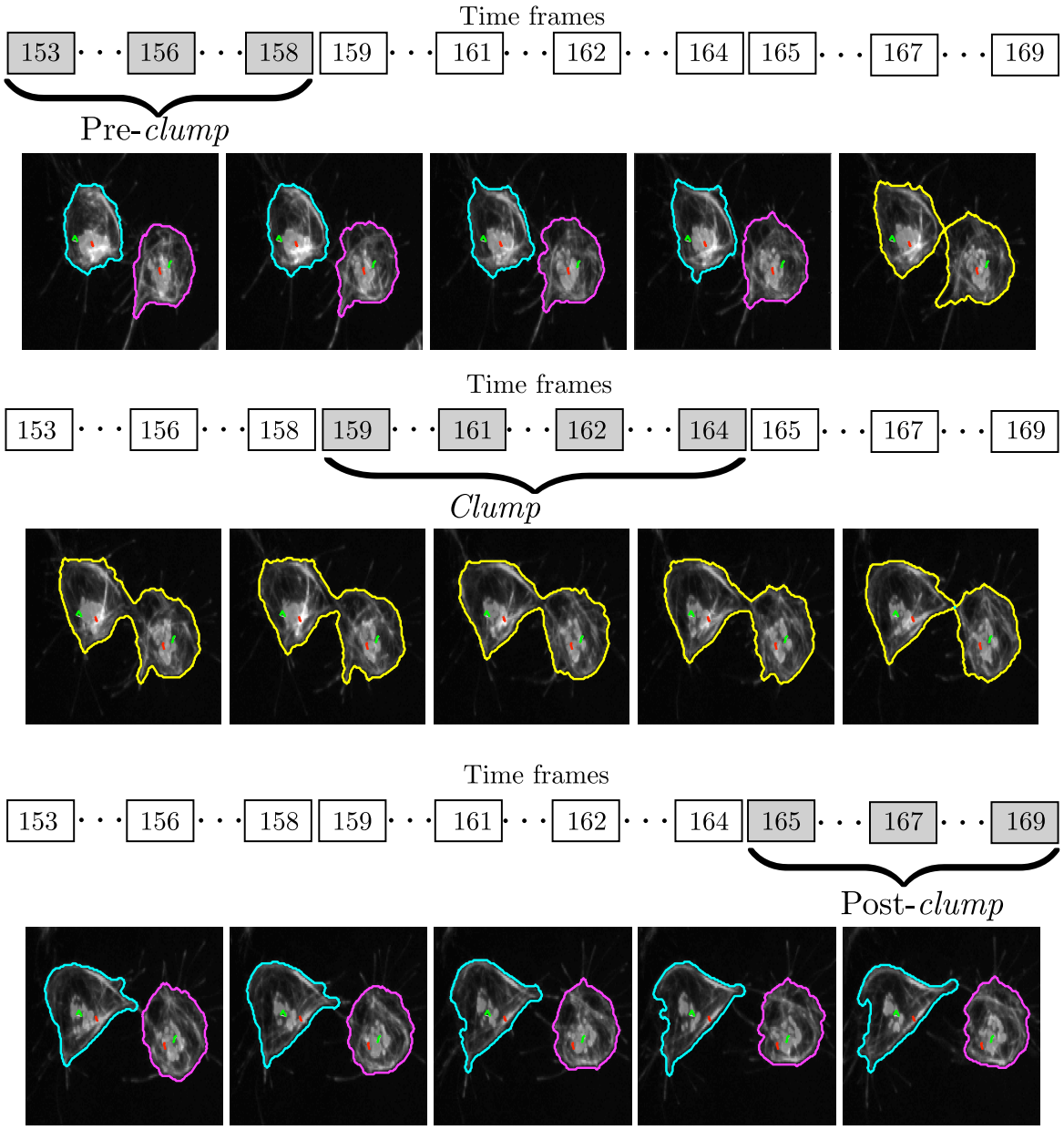


Figure 6.16 Representation of two cells in clump `2001` through the direction change and interaction analysis. The reader is referred to figure 5.8 in Chapter 5. The detected clump outlines are presented in yellow line and the individual cells are shown in cyan and magenta.

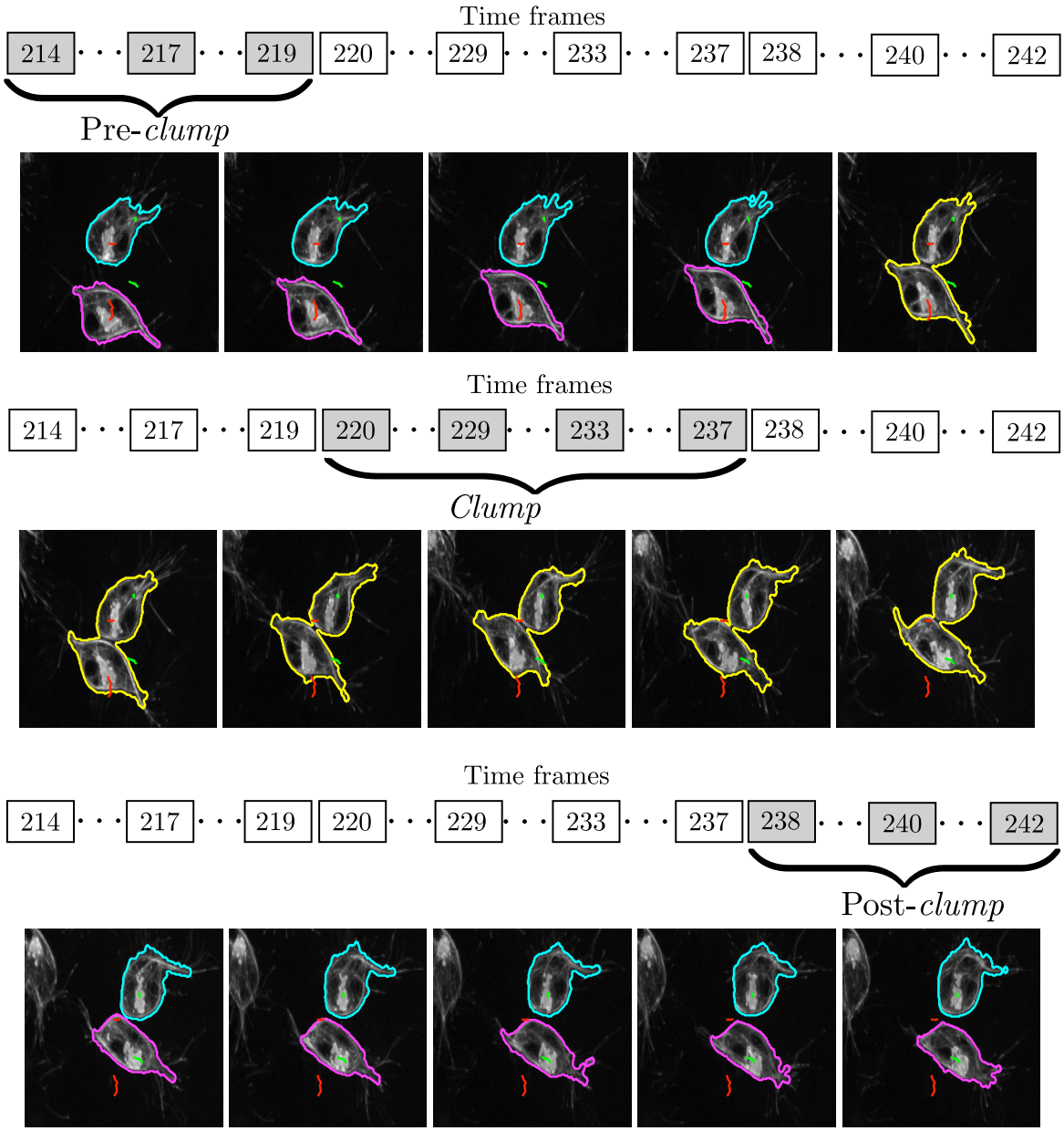


Figure 6.17 Representation of two cells in clump `3002` through the direction change and interaction analysis. The reader is referred to figure 5.8 in Chapter 5. The detected clump outlines are presented in yellow line and the individual cells are shown in cyan and magenta.

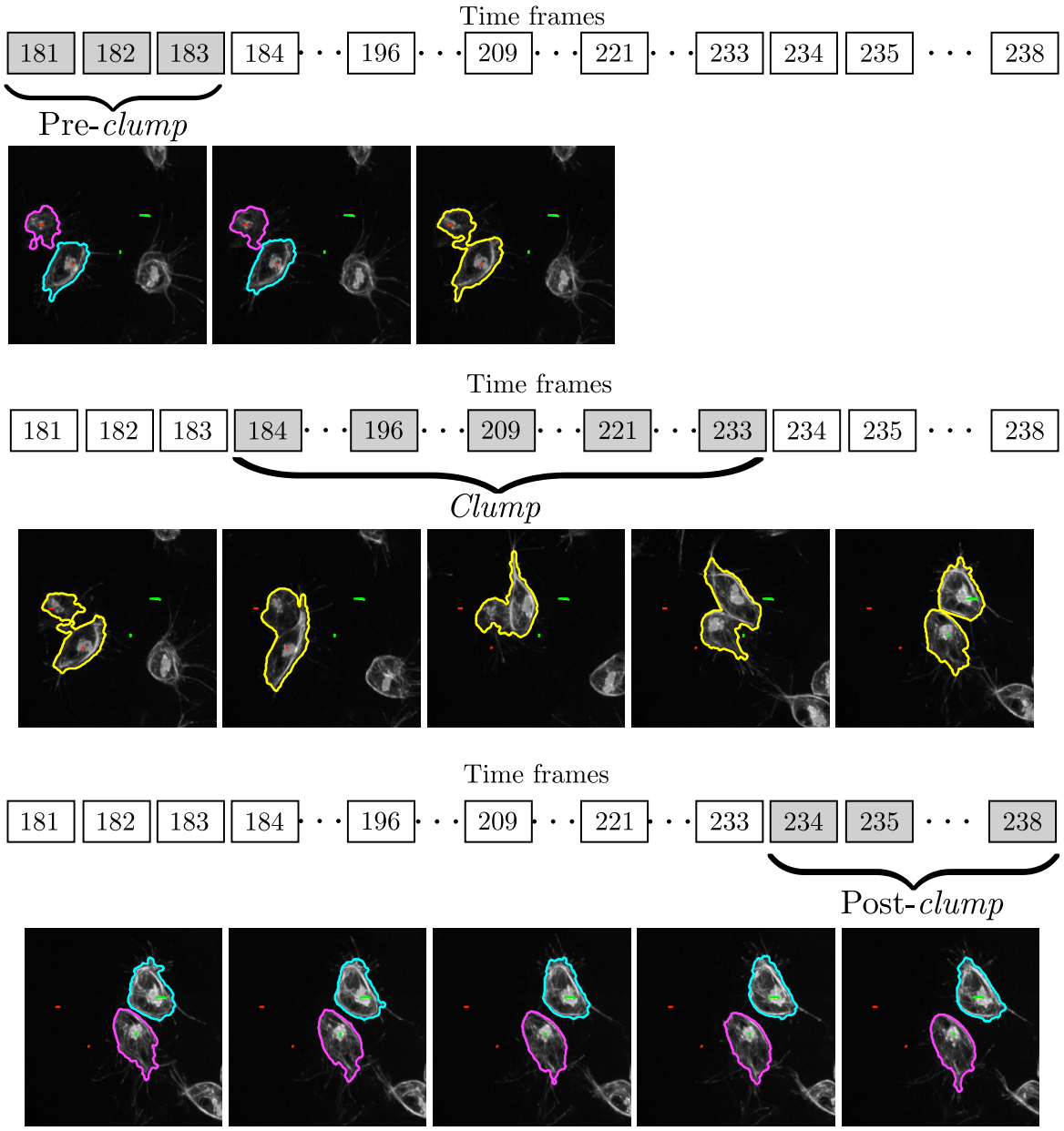
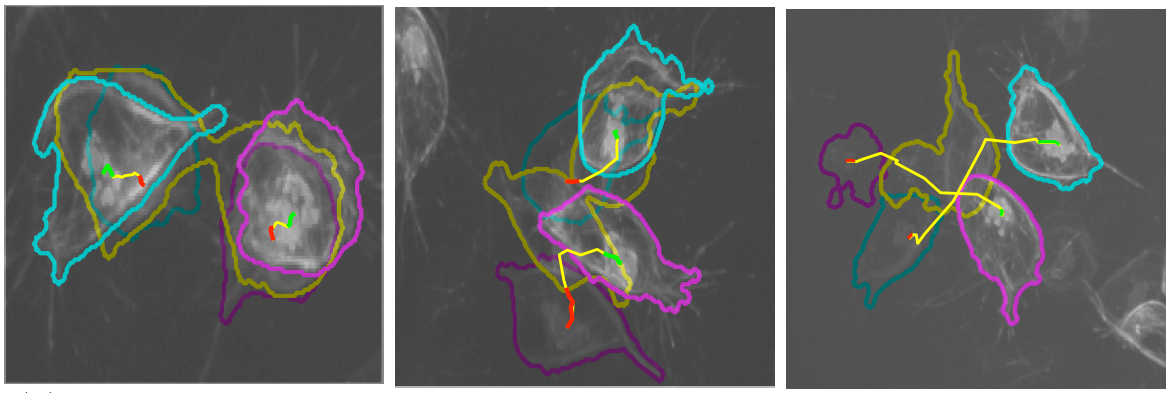


Figure 6.18 Representation of two cells in clump `22001` through the direction change and interaction analysis. The reader is referred to figure 5.8 in Chapter 5. The detected clump outlines are presented in yellow line and the individual cells are shown in cyan and magenta.



(a) Clump code: `2001` (b) Clump code: `3002` (c) Clump code: `22001`

Figure 6.19 Frames in clump interaction overlapped to appreciate cell movement and time in clump. In each case, three frames are superimposed: the first, middle and final frames in each experiment are shown, with the corresponding segmentations and tracks. The full track in each experiment is presented, with changes of colour representing different moments in the experiment: red - before the clump, yellow - during the clump and green - after the clump.

6.5 Summary

Results for all developments described in this thesis were presented in this chapter. The chapter was subdivided in three main projects: (i) the ISBI Cell Tracking Challenge of 2015 [64], (ii) the movement analysis of the *Artemia Franciscana* [53], (iii) comparison of the peak detection via *anglegram* and (iv) the developments described in the `macrosight` framework from chapter 5 and [108, 109]. Thus, this section is subdivided in a summary and discussion of points (i-iii) and comments on the developments in (iv).

Summary on additional tests

Discussion on ISBI results. The automatic thresholds found from the optimisation process described in equation (C.2), appear to have an overall better result other thresholds tested. Table 6.1 presenting the the automatic thresholds to have better results or close, next to the manually selected ones. However, the segmentation results in figure 6.4 are not optimal. Noticeable errors in the segmentation are appreciated in the datasets **C3DL-MDA**(d), **N3DH-CE**(i) and **N3DL-DRO**(m). It is worth noting that the datasets presenting the worse segmentation times were the ones that proved computationally expensive to test. Thus, the implementation of an optimisation technique for such datasets was made blindly given the promising outputs from the

2D datasets (table 6.1). Verifying the resulting segmented cells manually would have helped detect such problems. Subtle errors in segmentation can also be observed in noisy datasets like **C2DL-MSC**(b), **N2DH-GOWT1**(e) and **N3DH-CHO**(j), where the errors correspond to limitations in thresholding of the datasets.

Regarding tracking results, the most noticeable problems can be observed in datasets **N3DH-CHO**(j), where tracks appear to make large leaps from one point to another. This could be caused by a poor segmentation where a cell disappears from one volume to the next, and the keyhole algorithm assigning the corresponding track to another cell. Most notably, dataset **N3DH-CHO**(j) shows a single track.

The ISBI Cell Tracking Challenge was as an introductory project to the cell migration problem in image analysis. The challenge's results showed overall good tracking results, providing a suboptimal segmentation in many cases. This caused the subsequent projects, especially the work with macrophages to be heavily focused on segmentation, as the tracking solution was robust for many cases.

Discussion of the movement analysis of the *Artemia Franciscana*. The semi-automatic approach of studying the movement of the animals enabled to both automate and significantly increase sensitivity of *Artemia* toxicity biotests as compared to a current gold standard that is based on manual counting of immobile larvae at 24-hour end-point [53]. The approach is semi-automatic, with post-processing needed for the tracks, to join inconsistencies created by the incorrect segmentation of artefacts.

While a fully automatic approach would be desirable, the solution presented provide significant insights when analysing the moving patterns of the crustaceans. The reported results showed a similar trend as manually assessing the reduction of movement, albeit with differences towards higher concentrations of the pollutant. The differences were due to the software not taking into consideration of subtle movements such as the swing of an arm or agitation of the body.

Discussion of detection of peaks comparison to Harris' algorithm. The functionalities of the anglegram were compared with the Harris junction detector. Such a comparison is an extension from the original results reported in [108], and reproduced in Table 4.2. The results show improved accuracy in detecting circle shapes, compared to Table 4.2 due to the additional step when detecting more than four peaks.

Referring to Tables 6.2 and 6.3, a method was developed to obtain peaks using the Harris algorithm [103]. The method exploited the Harris' metric of the detected

junctions and selected those with a value above a certain threshold to have comparable results to the anglegram method. The peak detection via anglegram produced better results for all shapes identified, even when compared to a best-case-scenario alternative with the Harris' method (table 6.3).

Migrating macrophages analysed with `macrosight`.

In Chapter 5, the `macrosight` framework was described and its results presented in this section. Two functionalities were explored: **shape tracking and analysis** and **interactions analysis**. Three datasets were analysed: `MACROS1`, `MACROS2` and `MACROS3`, acquired with the same methodology described in [4, 43]. Each frame contains two channels, corresponding to the nuclei (red) and the microtubular structures (green). All methods assumed the baseline segmentation, described in chapter 4, and utilised the tracks obtained by PhagoSight.

Datasets `MACROS2` and `MACROS3` appear to have more contrast compared to `MACROS1`, which could explain the number of inconsistent tracks in it, as marked with black arrows in figure 6.9.

Discussion of shape evolution experiments. Shape evolution experiments were run on the `MACROS1` dataset, where all non-overlapping track fragments were isolated. Three tracks were selected to showcase the functionality of the shape tracking algorithm. This algorithm would allow to incorporate shape measurements to the track information, initially obtained by PhagoSight, including the location of peaks detected in the analysis.

The main results in this section are presented in figure 6.11. The three cases presented were chosen for each cell's shape changing and sizes; the experiments shown will be referred by the indexes in Figure 6.11, namely (a), (b) and (c). Problems when detecting peaks in the data correspond to errors in the peak detection via anglegram method. Such is the case of the first, fifth and seventh frames in (a.i). Other cases show two very close peaks being detected, like in most of the frames shown in (b.i), which could be attributed to the segmentation of the green channel. A postprocessing method to the anglegram could take such cases and join the segmentations into a single peak. Also, in (a.ii) the orientation angles towards the final frames show a large transition caused by warping of the angles.

As mentioned in [108], the key contribution is the consistent tracking of the shape providing a measured evolution of shape parameters. More examples have been showcased which further such statement. Furthermore, this implementation could

incorporate such measurements as information in the tracks computed by PhagoSight. Minor errors can be addressed through postprocessing and by a thorough understanding of the anglegram.

Discussion of cell direction changes. Interactions analysis experiments were run on all `MACROS` datasets, exploiting the consistency in the intensities of the images to obtain reliable tracks. Thirty seven experiments were run in total, which would fulfil the criteria described before. It is worth noting that the term *interaction* is used to describe two cells coming in contact to a clump. Also, even though only two cells came in contact in a given clump, the time spent in it varied widely, as seen in Figure 6.15(a).

Consider the comparisons of the changes in track directions with or without clump. Figure 6.12 represents all tracks rotated and translated to coincide at a specific point. It is worth noting that for the single cell movement experiments, in Figure 6.12(ii), the reference point was chosen arbitrarily as the middle point in the track segment leading up to the clump. Furthermore, Figure 6.12 presents more insights about the overall hypothesis presented in this thesis. For [109] the analysis was only performed in the `MACROS3` dataset, shown in Figure 6.12(c), where a clear difference in the tracks can be observed, although the angle variation could not be proven to be different with statistical significance.

Segments of this discussion are reproduced verbatim from the work presented in [113]. Figure 6.12 shows tracks to be different between the two cases. The cases of cell-cell contact show less movement before and after contact, shown by the smaller lines; a higher variability is observed in control cases. Each dataset also presents unique differences in the comparisons. In `MACROS2`, for example, the tracks appear more curved after the interaction; and in `MACROS3`, the ranges of change in direction are more distinct.

Even though the differences between contact and control cases are noticeable, however the characterisation of the movement should not be reduced to the analysis of a single variable, in this case the angle of changed direction. For example, the tracks corresponding to interactions travel very short distances with curved movements, causing the calculation of the angle variations to be much smaller. The tracks corresponding to single cell movement seem much more chaotic. Upon close inspection of each track, it was observed how some of the tracks before a clump took a turn towards it in the 11 frames leading up to the interaction. In both cases, the movement pattern do appear

different in one case from another, however it seems to be a complex process where the basic angle change measurements were insufficient to categorise the movement. More variables like speed, curvature could be incorporated. From a biological standpoint, the moment cells come in contact could be determined much sooner if the microtubules with a lower intensities are also segmented and tracked.

An extension to the results reported in [109] was presented. The number of time frames a cell remained in a clump **time in clump** TC, was presented to try and contextualise the differences in motion patterns in the cells. The hypothesis being that the time each cell spent in each clump would impact the angle variation. Figures 6.14, and subsequently 6.15 were created to observe the angle differences per dataset, given the time in clump. The time in clump, TC, was found to be a relevant parameter for the measured angle change for different reasons. Firstly, the value of TC per clump appeared to be smaller where statistical significance was achieved, as seen in table 6.4. Secondly, from figures 6.14 and 6.15 display the differences in the angle changes and their ranges if the experiments are restricted to keeping a low time in clump ($TC < 10$). Finally, as mentioned in Section 2.2, each frame is taken every 10 seconds, and the time cells remain in contact should not exceed 1-2 minutes ($TC \in [6, 12]$) to be considered an interaction relevant to explore. Through the analysis made in this work, the time in clump was found to be helpful when assessing the change of direction in values of $TC < 10$. The result was not previously found in [109], therefore it fuels the need for a more thorough analysis of the tracks' parameters, attempting to characterise them.

In both datasets, a lot of cases seem to be grouped at specific values of TC. For example, in Figure 6.15(d) where a purple line was drawn at $TC = 10$, showing six cases of different angle changes. To investigate further, all experiments were partitioned based on a specific time in clump in cases with $TC \leq a$ and $TC > a$, $a = 2, 4, 6, 10$. The results were shown in Figure 6.15, where the angle change distributions of each experiment vary noticeably depending on the values of time in clump considered. The rightmost column is noticeable, as it is the one involving pairs of cells which interact in what would be described as brief time periods, it also presents the most apparent differences in the distributions of the angle changes between experiments with clumps and single cell movement experiments. Finally, three experiments were chosen with different time in clump, so the movement in the cells can be observed in the context of the images. In figures 6.16, 6.17 and 6.18, an overview of the entire movement is shown.

The limitations of the algorithm involved the underlying limitations of the segmentation and tracking methods at dealing with complicated interactions. As mentioned before, the experiments were chosen manually, by observing the *clump span* of each case. Upon verification of the tracks, which consists of manually comparing the segmented nuclei and tracks, the dataset `MACROS1` presented some inconsistencies like nuclei changing their track identifier, thus complicating the choosing of experiments. The problems with the tracks could be inherited from the limitations of the keyhole algorithm, which only considers velocity and previous direction to estimate the upcoming position. The reader is referred to figure 6.12(b.i), where the red tracks appear clearly curved. The problems could also be due to segmentation, as `MACROS1` contains more variability in the intensities than `MACROS2` and `MACROS3`, as well as a larger number of cells interacting in each clump. Future work could improve this by incorporating post-processing to the tracks.

Chapter 7

Conclusions and Future Work

A framework for the analysis of moving objects, with special emphasis on fluorescently labelled macrophages through image analysis and tracking methods has been presented. The framework includes the analysis of the segmentation, tracking and shape analysis. The framework was developed to analyse the movement and provide insights on the *intention* of movement from migrating cells, shedding light on the understanding of the processes that drive cell movement. This chapter presents a summary of the thesis, which highlights the key findings; then, the major contributions and conclusions; finally, limitations of the algorithms are presented, leading to suggestions for further research.

7.1 Summary

Cell migration as a biological process, the mathematical modelling to describe it and the acquisition of images and time sequences through microscopy were first described to provide the context to the data studied. Cell migration is an intricate process involving physical and chemical interactions within the context of the biological structures of cells and the substrate. Despite the comprehensive knowledge about the movement of cells, the ways in which cells integrate cues is not yet fully understood. Macrophages were presented as an example of cells whose migratory functions are of great importance to the health of organisms. The data of macrophages was presented and described in detail as images with two levels of fluorescence producing two separate layers: nuclei in red and microtubules in green.

To study the movement of macrophages present in the data, the images were thoroughly explored by several image analysis techniques. Chapter 3 was then used both as literature review and as an exploration of the data. Classic image analysis

techniques were explored including preprocessing, segmentation by thresholding and postprocessing. Different methodologies were used to provide a thorough exploration of the data. More sophisticated techniques, such as a multilevel set method and active shape models, were described and tested on the data with unsatisfactory results that would not justify the increase in complexity compared to results achieved by simpler techniques.

A baseline segmentation methodology was developed to segment objects in frames where intensities are distributed similarly to fluorescence microscopy frames. Such segmentation was aimed to detect the foreground in both channels of the data, more importantly, it would allow for the detection of overlapped cells in the green channel, *clumps*.

The framework for the analysis of moving macrophages was divided into the segmentation and tracking. At every segmented, analysis of the boundaries of the detected objects was performed through a novel technique based on a multiscale analysis of the inner angles of the boundary of an object, called *anglegram* matrix. The algorithm allowed for the detection of junctions, whether concave (corners) or convex (peaks), through the detection of maxima in a projection of the anglegram's rows. Two applications for detecting junctions with the anglegram were presented: (i) disambiguation of clumps and (ii) analysis of shape through detection of peaks. In both cases, the anglegram method was compared to synthetic data, outperforming the Harris corner detector [103]. In overlapping cells, corner junctions in clumps were used as points where the underlying boundaries of the cells intersect; for it, a proposal of exploiting the junctions was presented. In single cells, movement is accompanied by shape changes. Four shapes were identified from the data, depending on the number of peaks; junctions detected by the anglegram were used to identify the shapes as cells migrated.

The full framework, `macrosight`, analysed the movement of cells as they migrate. The framework implemented the PhagoSight's keyhole tracking algorithm [102] to link the segmented nuclei between frames. Two algorithms were presented: (i) shape tracking, to acquire the shape information from the segmented green channel on non overlapping cells; (ii) analysis of interactions, which compared the general orientation of movement before and after a cell came in contact with another one, forming a clump. Results presented for the shape tracking analysis show a consistent tracking providing a measured evolution of shape parameters.

7.2 Major contributions

- A thorough exploration of the data obtained in the work by Stramer et al. [4], throughout Chapter 3 and supported by the context of cell migration and imaging techniques presented in chapter 2. Classic image analysis techniques were reviewed as an exploration of the data, showing the difficulties in segmentation, not including overlapping cases and a classical segmentation was proposed to segment both channels.

- To analyse the shapes of the detected objects, the *anglegram* matrix was proposed to detect important junctions in the boundaries of objects. The anglegram was validated with synthetic data for both types of junctions. More importantly, the algorithm detected junctions comparable to the Harris corner detector but with the added benefit of detecting junctions based on context, allowing the distinction between peak and corner junctions. In detection of peaks, the algorithm showed improved results over Harris' method. Applications of the anglegram included the identification of shapes based on the number of peaks detected on single cells and the use of corner junctions for the disambiguation of clumps; showing promising results were found on both applications.

- The `macrosight` framework was proposed for the analysis of moving macrophages, providing the separate tools to do so. It includes a robust segmentation for both channels. The framework incorporates an established tracking technique for the nuclei (red channel) to the shapes of single cells or clumps in the green channel.

- Throughout the development of this thesis, algorithms were developed in MATLAB® and versioned using git. The following projects including the most important algorithms in this thesis are available on Github, an online repository for sharing code. Anglegram, described in chapter 4[1]. The `macrosight` framework, described in chapter 5[2], and the manual segmentation of time sequence frames, described in chapter 2[3].

- The algorithms presented in this work can be extended into other applications, as presented on the ISBI and *Artemia* projects.

[1]https://github.com/alonsoJASL/matlab.anglegram
[2]https://github.com/alonsoJASL/macrosight
[3]https://github.com/alonsoJASL/matlab.manualSegmentation

7.3 Major conclusions

The following conclusions are considered:

- The macrophages time sequences analysed in this thesis are an important development in imaging techniques, allowing for the observation of microtubular structures, and cell interactions as not available before. However, the amount of structural information of the cells migrating and interacting, have created a remarkably complex dataset, with several challenges when analysing it with computer algorithms. Being a novel dataset, annotated ground truth was not available to explore supervised machine learning approaches. The cells adopt extremely varied shapes, such that other methodologies with strict shape assumptions like the level set method by Lu et al. [99], the work by Plissiti and Nikou [96] or even active shape models explored in Chapter 3, could not work with the data.

- One key hypothesis raised in this thesis, as observed in the data, was the possibility that the interactions between the cells could affect their moving patterns. This hypothesis was explored through the `macrosight` framework, and its results presented in section 6.4. The comparison of the change angles before and after the interaction resulted to be valuable, but insufficient as there was no significant difference. Yet visually, the tracks appear to be different, which invites further analysis. A key conclusion from this analysis is worth noting: the test was done on a single variable of movement, *i.e.* angle, and movement led by interactions was assumed to influence it. More variables such as speed, curvature of the track, polarity of the cell, and shape parameters, including cell state should be considered in the analysis to produce a comprehensive description of the movement.

- Following the previous point, the shape analysis of the macrophages provide an interesting way to contextualise the nuclei movement to the shape measurements of the cells. It has been observed in the data, how cells appear to elongate as they migrate towards another cell or clump. In some cases, the shape change is evident, from a circle into a drop if only another single cell is close to it; or a bidrop or tridrop if more cells are present in the vicinity. A calculation of the basic shapes described in Section 5.2 allowed an initial approach to defining states in cellular movement, based on their interactions that have not been explored.

7.4 Limitations

The limitations presented include all major blocks of the `macrosight` framework: the segmentation and anglegram from Chapter 4, as well as the shape tracking and the influence of cell-cell contact in driving the movement of cells. The limitations are highlighted to introduce possible routes moving the research forward.

Baseline segmentation. Several microtubular structures are removed from the analysis with the morphological operations, causing certain events of contact to go undetected. The need for a closed *almost* convex curve causes some clumps to be more complicated than need be. As an example, the reader is referred to figure 4.4.

Anglegram and junction detection. The anglegram is completely dependant on the previous segmentation step, and has no way of adjusting itself using other information, such as the intensities. Such adjustment, done *a posteriori*, could also help produce a more accurate map of the junctions, even in cases where the angles measured in the anglegram are not high enough to be detected.

Shape tracking analysis. The main limitation of this technique is that it is not robust enough to address overlapping cases. Some restrictions on the active contour's evolution might address the leaking problem. In the **evolve** function, the different functionalities (shrink, grow and normal) were implemented to address this, however they do not follow a precise measurement the intensities around the boundary of the object, instead they only assume the area of the objects should remain relatively constant. More restrictions on the boundaries' shapes could aid this problem.

Analysis of interactions. As this analysis is the last in the pipeline, the limitations on previous algorithms would impact the performance of it, those are not mentioned. However, other limitations can be found independent of the methodologies this method depends on: (i) not being able to distinguish between types of contact, e.g. actual interaction vs. overlapping due to the cells being in different planes, in such cases, an expert would need to label each contact to determine interactions; (ii) the algorithm is semi-automatic, with the cases needing to be manually written logged in a spreadsheet; (iii) as mentioned before, a detected *clump* does not mean a contact event, however this is the case in the current implementation of the work.

7.5 Suggestions for further research

The following suggestions for research do not necessarily address the problems stated in the previous section, but provide further extensions of the work.

- As mentioned before, the frames in each dataset summarise various information at different scales. In this thesis, an exploration was made on the general shape of the cell when migrating, however, when observing the data, several microtubular structures (pseudopodia) are observed, which were intentionally removed from the analysis. Having a robust segmentation and shape analysis, the research can focus on the smaller structures observed incorporating them into the analysis.

- The anglegram is a noteworthy tool for the analysis of shape, and several improvements could be considered. First, the anglegram could be exploited as a prior to a probabilistic modelling of the position of the junctions and the overall shape. This would allow to deal with complex shapes, where corner junctions are missed because the inner angles are not high enough. Using the anglegram as a shape prior could provide insights to recovering a shape from the anglegram.

- Following the previous point, the dimension of the anglegram could be extended to consider grey level images, and a likely shape, similar to a Hough transform.

- A clear future direction of this research would expand the notion of tracks to include shape parameters and spatial context to identify interactions of cells. Extracting features from a track and categorise it is an idea explored before [114] and incorporated to the information already obtained in this work could produce satisfactory results.

- Cells seem to elongate when approaching other cells, thus an analysis of the spatial context of the cells to the rest of the cells could be explored. Quantification of such elongation from a rounded circle into basic shapes was introduced with the *pointiness* of the shape and could provide more information to the overall description of a moving track. A noteworthy point when designing a more thorough measurement of the characteristics of movement is subject such technique to thorough validation using synthetic and real data.

Closing statement. Cell migration was presented in this work as a complex biological problem involving a two complementary approaches to understanding it: image analysis or mathematical modelling. This work analysed the problem from the perspective of image analysis because the interactions being measured have not been fully understood. Through the analysis presented in this work, the determination of the interactions presented in this work might be useful in the modelling of whole cell dynamics, providing insights of motion dynamics, which paired to shape states, have not been fully accounted before. The evolution of shapes before a contact event would be relevant to mathematical models. Finally, the notion of a *shape signature* was explored with the anglegram, which could aid in geometrical image analysis.

The work presented in this thesis lays the foundation for automated image analysis of migrating macrophages. Several studies of cell motion are still analysed manually, especially when it comes to segmentation. The work presented in this thesis provides a framework which would allow biologists to have consistent measurements of the phenomena in the movies, thus allowing for reproducibility in studies with more complex questions, such as determining the differences between control cells vs. cells in which a specific gene has been modified. The framework could be extended to incorporate a more detailed analysis of the cells' tracks incorporating velocities and shape evolution. The amount of information scientists acquire is vast, therefore an automated analysis would highlight patterns in the data not seen before. The project presented in this thesis follows a very specific scope, however it provides clear implications in several fields of research.

References

[1] Jens Rittscher, Raghu Machiraju, and Stephen T. C. Wong. *Microscopic Image Analysis for Life Science Applications*. Artech House, Inc., July 2008.

[2] Fernando Oneissi Martinez, Antonio Sica, Alberto Mantovani, and Massimo Locati. Macrophage activation and polarization. *Frontiers in Bioscience: A Journal and Virtual Library*, 13:453–461, January 2008.

[3] Will Wood and Paul Martin. Macrophage Functions in Tissue Patterning and Disease: New Insights from the Fly. *Developmental Cell*, 40(3):221–233, February 2017.

[4] Brian Stramer, Severina Moreira, Tom Millard, Iwan Evans, Chieh-Yin Huang, Ola Sabet, Martin Milner, Graham Dunn, Paul Martin, and Will Wood. Clasp-mediated microtubule bundling regulates persistent motility and contact repulsion in Drosophila macrophages in vivo. *Journal of Cell Biology*, 189(4):681–689, May 2010.

[5] Shirin M. Pocha and Denise J. Montell. Cellular and molecular mechanisms of single and collective cell migrations in Drosophila: themes and variations. *Annual Review of Genetics*, 48:295–318, 2014.

[6] Ryan J Petrie, Andrew D Doyle, and Kenneth M Yamada. Random versus directionally persistent cell migration. *Nature reviews. Molecular cell biology*, 10 (8):538–49, August 2009.

[7] Isabella Ellinger and Adolf Ellinger. Smallest Unit of Life: Cell Biology. In Erika Jensen-Jarolim, editor, *Comparative Medicine: Anatomy and Physiology*, pages 19–33. Springer Vienna, Vienna, 2014. ISBN 978-3-7091-1559-6.

[8] David G. Drubin and W. James Nelson. Origins of Cell Polarity. *Cell*, 84(3): 335–344, February 1996.

[9] G. E. Jones. Cellular signaling in macrophage migration and chemotaxis. *Journal of Leukocyte Biology*, 68(5):593–602, November 2000.

[10] Douglas A. Lauffenburger and Alan F. Horwitz. Cell Migration: A Physically Integrated Molecular Process. *Cell*, 84(3):359–369, February 1996.

[11] Anne J. Ridley, Martin A. Schwartz, Keith Burridge, Richard A. Firtel, Mark H. Ginsberg, Gary Borisy, J. Thomas Parsons, and Alan Rick Horwitz. Cell Migration: Integrating Signals from Front to Back. *Science*, 302:1704–1709, December 2003.

[12] Rick Horwitz and Donna Webb. Cell migration. *Current Biology*, 13(19):R756–759, September 2003.

[13] Carole A. Parent and Peter N. Devreotes. A Cell's Sense of Direction. *Science*, 284(5415):765–770, April 1999.

[14] Chun-Min Lo, Hong-Bei Wang, Micah Dembo, and Yu-li Wang. Cell Movement Is Guided by the Rigidity of the Substrate. *Biophysical Journal*, 79(1):144–152, July 2000.

[15] Min Zhao. Electrical fields in wound healing—An overriding signal that directs cell migration. *Seminars in Cell & Developmental Biology*, 20(6):674–682, August 2009.

[16] Peter J. M. Van Haastert and Peter N. Devreotes. Chemotaxis: signalling the way forward. *Nature Reviews. Molecular Cell Biology*, 5(8):626–634, August 2004.

[17] Visakan Kadirkamanathan, Sean R. Anderson, Stephen A. Billings, Xiliang Zhang, Geoffrey R. Holmes, Constantino C. Reyes-Aldasoro, Philip M. Elks, and Stephen A. Renshaw. The Neutrophil's Eye-View: Inference and Visualisation of the Chemoattractant Field Driving Cell Chemotaxis In Vivo. *PLOS ONE*, 7(4): e35182, April 2012.

[18] Mark S. Bretscher. On the shape of migrating cells–a 'front-to-back' model. *Journal of Cell Science*, 121(Pt 16):2625–2628, August 2008.

[19] Armin Haupt and Nicolas Minc. How cells sense their own shape – mechanisms to probe cell geometry and their implications in cellular organization and function. *J Cell Sci*, 131(6):jcs214015, March 2018.

[20] Ajay Chawla, Jeffrey W. Pollard, and Thomas A. Wynn. Macrophage biology in development, homeostasis and disease. *Nature*, 496(7446):445, April 2013.

[21] Andrew D. Luster, Ronen Alon, and Ulrich H. von Andrian. Immune cell migration in inflammation: present and future therapeutic targets. *Nature Immunology*, 6(12):1182–1190, December 2005.

[22] Charles R. Mackay. Moving targets: cell migration inhibitors as new anti-inflammatory therapies. *Nature Immunology*, 9(9):988–998, September 2008.

[23] Brendan Flaherty, J. P. McGarry, and P. E. McHugh. Mathematical Models of Cell Motility. *Cell Biochemistry and Biophysics*, 49(1):14–28, September 2007.

[24] F. J. Vermolen. Short review of computational models for single cell deformation and migration. *Single Cell Biology*, 4(1):1–2, 2015.

[25] Steven Munevar, Yu-li Wang, and Micah Dembo. Traction Force Microscopy of Migrating Normal and H-ras Transformed 3t3 Fibroblasts. *Biophysical Journal*, 80(4):1744–1757, April 2001.

[26] Charles M. Elliott, Björn Stinner, and Chandrasekhar Venkataraman. Modelling cell motility and chemotaxis with evolving surface finite elements. *Journal of The Royal Society Interface*, page rsif20120276, June 2012.

[27] Wayne Croft, Charles M. Elliott, Graham Ladds, Björn Stinner, Chandrasekhar Venkataraman, and Cathryn Weston. Parameter identification problems in the modelling of cell motility. *Journal of Mathematical Biology*, 71(2):399–436, August 2015.

[28] Uduak Z. George, Angélique Stéphanou, and Anotida Madzvamuse. Mathematical modelling and numerical simulations of actin dynamics in the eukaryotic cell. *Journal of Mathematical Biology*, 66(3):547–593, February 2013.

[29] A. Mogilner and B. Rubinstein. The Physics of Filopodial Protrusion. *Biophysical Journal*, 89(2):782–795, August 2005.

[30] Erdinç Atilgan, Denis Wirtz, and Sean X. Sun. Morphology of the Lamellipodium and Organization of Actin Filaments at the Leading Edge of Crawling Cells. *Biophysical Journal*, 89(5):3589–3602, November 2005.

[31] Anders E. Carlsson. Growth of Branched Actin Networks against Obstacles. *Biophysical Journal*, 81(4):1907–1923, October 2001.

[32] A. Stéphanou, M. A. J. Chaplain, and P. Tracqui. A mathematical model for the dynamics of large membrane deformations of isolated fibroblasts. *Bulletin of Mathematical Biology*, 66(5):1119, September 2004.

[33] F. J. Vermolen and A. Gefen. A phenomenological model for chemico-mechanically induced cell shape changes during migration and cell–cell contacts. *Biomechanics and Modeling in Mechanobiology*, 12(2):301–323, April 2013.

[34] Anotida Madzvamuse and Uduak Zenas George. The moving grid finite element method applied to cell movement and deformation. *Finite Elements in Analysis and Design*, 74:76–92, October 2013.

[35] Feng Wei Yang, Chandrasekhar Venkataraman, Vanessa Styles, Verena Kuttenberger, Elias Horn, Zeno von Guttenberg, and Anotida Madzvamuse. A computational framework for particle and whole cell tracking applied to a real biological dataset. *Journal of Biomechanics*, 49(8):1290–1304, May 2016.

[36] J.P. Hornak. *Encyclopedia of Imaging Science and Technology, 2 Volume Set.* Wiley, 2002. ISBN 9780471332763.

[37] Dirk Dormann and Cornelis J. Weijer. Imaging of cell migration. *The EMBO journal*, 25(15):3480–3493, August 2006.

[38] Joost B. Beltman, Athanasius F. M. Marée, and Rob J. de Boer. Analysing immune cell migration. *Nature Reviews. Immunology*, 9(11):789–798, November 2009.

[39] Ulrich Kubitscheck. *Fluorescence Microscopy - From Principles to Biological Applications (2nd Edition).* John Wiley & Sons, 2017.

[40] Richard P Haugland. *The handbook: a guide to fluorescent probes and labeling technologies*, volume 10. Invitrogen Molecular probes, 2005.

[41] Marc Zimmer. Green Fluorescent Protein (GFP): Applications, Structure, and Related Photophysical Behavior. *Chemical Reviews*, 102(3):759–782, March 2002.

[42] S James Remington. Fluorescent proteins: maturation, photochemistry and photophysics. *Current Opinion in Structural Biology*, 16(6):714–721, December 2006.

[43] Brian Stramer and Will Wood. Inflammation and Wound Healing in Drosophila. In *Chemotaxis*, Methods in Molecular Biology™, pages 137–149. Humana Press, Totowa, NJ, 2009.

[44] Graham J. Lieschke and Peter D. Currie. Animal models of human disease: zebrafish swim into view. *Nature Reviews Genetics*, 8(5):353–367, May 2007.

[45] Peter Friedl and Darren Gilmour. Collective cell migration in morphogenesis, regeneration and cancer. *Nature Reviews. Molecular Cell Biology*, 10(7):445–457, July 2009.

[46] Luanne L. Peters, Raymond F. Robledo, Carol J. Bult, Gary A. Churchill, Beverly J. Paigen, and Karen L. Svenson. The mouse as a model for human biology: a resource guide for complex trait analysis. *Nature Reviews Genetics*, 8 (1):58–69, January 2007.

[47] Michael Gerharz, Ankev Baranowsky, Udo Siebolts, Sabine Eming, Roswitha Nischt, Thomas Krieg, and Claudia Wickenhauser. Morphometric analysis of murine skin wound healing: Standardization of experimental procedures and impact of an advanced multitissue array technique. *Wound Repair and Regeneration*, 15(1):105–112, January 2007.

[48] Maura C. Kibbey, Derrick S. Grant, and Hynda K. Kleinman. Role of the SIKVAV Site of Laminin in Promotion of Angiogenesis and Tumor Growth: An In Vivo Matrigel Model. *JNCI: Journal of the National Cancer Institute*, 84(21): 1633–1638, November 1992.

[49] Stephanie Alexander, Gudrun E. Koehl, Markus Hirschberg, Edward K. Geissler, and Peter Friedl. Dynamic imaging of cancer growth and invasion: a modified skin-fold chamber model. *Histochemistry and Cell Biology*, 130(6):1147–1154, December 2008.

[50] Constantino Carlos Reyes-Aldasoro, Meit Bjorndahl, Chryso Kanthou, and Gillian M. Tozer. Topological Analysis of the Vasculature of Angiopoietin-Expressing Tumours Through Scale-Space Tracing. In *Medical Image Understanding and Analysis*, Communications in Computer and Information Science, pages 285–296. Springer, Cham, July 2017.

[51] Guido Persoone and Peter G Wells. Artemia in aquatic toxicology: a review. *Artemia research and its applications*, 1:259–275, 1987.

[52] L. Lewan, M. Andersson, and P. Morales-Gomez. Use of Artemia salina in toxicity testing. *Alternatives to laboratory animals: ATLA*, 1992.

[53] Jose Alonso Solis-Lemus, Yushi Huang, Donald Wlodkowic, and Constantino Carlos Reyes-Aldasoro. Microfluidic environment and tracking analysis for the observation of Artemia Franciscana. pages 4.1–4.8. British Machine Vision Association, 2015.

[54] Genetic Science Learning Center. Brine shrimp life cycle. URL https://learn.genetics.utah.edu/content/gsl/artemia/.

[55] Markus Affolter, Savério Bellusci, Nobuyuki Itoh, Benny Shilo, Jean-Paul Thiery, and Zena Werb. Tube or Not Tube: Remodeling Epithelial Tissues by Branching Morphogenesis. *Developmental Cell*, 4(1):11–18, January 2003.

[56] Markus Affolter and Emmanuel Caussinus. Tracheal branching morphogenesis in Drosophila: new insights into cell behaviour and organ architecture. *Development*, 135(12):2055–2064, June 2008.

[57] Andy Aman and Tatjana Piotrowski. Wnt/β-Catenin and Fgf Signaling Control Collective Cell Migration by Restricting Chemokine Receptor Expression. *Developmental Cell*, 15(5):749–761, November 2008.

[58] Petra Haas and Darren Gilmour. Chemokine Signaling Mediates Self-Organizing Tissue Migration in the Zebrafish Lateral Line. *Developmental Cell*, 10(5): 673–680, May 2006.

[59] C. C. Reyes-Aldasoro, Y. Zhao, D. Coca, S. A. Billings, V. Kadirkamanathan, G. M. Tozer, and S. A. Renshaw. Analysis of immune cell function using in vivo cell shape analysis and tracking. *4th IAPR International Conference on Pattern Recognition in Bioinformatics*, May 2009.

[60] Mark D. Adams, Susan E. Celniker, Robert A. Holt, Cheryl A. Evans, Jeannine D. Gocayne, Peter G. Amanatides, Steven E. Scherer, Peter W. Li, Roger A. Hoskins, Richard F. Galle, Reed A. George, Suzanna E. Lewis, Stephen Richards, Michael Ashburner, Scott N. Henderson, Granger G. Sutton, Jennifer R. Wortman, Mark D. Yandell, Qing Zhang, Lin X. Chen, Rhonda C. Brandon, Yu-Hui C. Rogers, Robert G. Blazej, Mark Champe, Barret D. Pfeiffer, Kenneth H. Wan, Clare Doyle, Evan G. Baxter, Gregg Helt, Catherine R. Nelson, George L. Gabor, Miklos, Josep F. Abril, Anna Agbayani, Hui-Jin An, Cynthia Andrews-Pfannkoch, Danita Baldwin, Richard M. Ballew, Anand Basu, James Baxendale, Leyla Bayraktaroglu, Ellen M. Beasley, Karen Y. Beeson, P. V. Benos, Benjamin P. Berman, Deepali Bhandari, Slava Bolshakov, Dana Borkova, Michael R. Botchan, John Bouck, Peter Brokstein, Phillipe Brottier, Kenneth C. Burtis, Dana A. Busam, Heather Butler, Edouard Cadieu, Angela Center, Ishwar Chandra, J. Michael Cherry, Simon Cawley, Carl Dahlke, Lionel B. Davenport, Peter Davies, Beatriz de Pablos, Arthur Delcher, Zuoming Deng, Anne Deslattes Mays, Ian Dew, Suzanne M. Dietz, Kristina Dodson, Lisa E. Doup, Michael Downes, Shannon Dugan-Rocha, Boris C. Dunkov, Patrick Dunn, Kenneth J. Durbin, Carlos C. Evangelista, Concepcion Ferraz, Steven Ferriera, Wolfgang Fleischmann,

Carl Fosler, Andrei E. Gabrielian, Neha S. Garg, William M. Gelbart, Ken Glasser, Anna Glodek, Fangcheng Gong, J. Harley Gorrell, Zhiping Gu, Ping Guan, Michael Harris, Nomi L. Harris, Damon Harvey, Thomas J. Heiman, Judith R. Hernandez, Jarrett Houck, Damon Hostin, Kathryn A. Houston, Timothy J. Howland, Ming-Hui Wei, Chinyere Ibegwam, Mena Jalali, Francis Kalush, Gary H. Karpen, Zhaoxi Ke, James A. Kennison, Karen A. Ketchum, Bruce E. Kimmel, Chinnappa D. Kodira, Cheryl Kraft, Saul Kravitz, David Kulp, Zhongwu Lai, Paul Lasko, Yiding Lei, Alexander A. Levitsky, Jiayin Li, Zhenya Li, Yong Liang, Xiaoying Lin, Xiangjun Liu, Bettina Mattei, Tina C. McIntosh, Michael P. McLeod, Duncan McPherson, Gennady Merkulov, Natalia V. Milshina, Clark Mobarry, Joe Morris, Ali Moshrefi, Stephen M. Mount, Mee Moy, Brian Murphy, Lee Murphy, Donna M. Muzny, David L. Nelson, David R. Nelson, Keith A. Nelson, Katherine Nixon, Deborah R. Nusskern, Joanne M. Pacleb, Michael Palazzolo, Gjange S. Pittman, Sue Pan, John Pollard, Vinita Puri, Martin G. Reese, Knut Reinert, Karin Remington, Robert D. C. Saunders, Frederick Scheeler, Hua Shen, Bixiang Christopher Shue, Inga Sidén-Kiamos, Michael Simpson, Marian P. Skupski, Tom Smith, Eugene Spier, Allan C. Spradling, Mark Stapleton, Renee Strong, Eric Sun, Robert Svirskas, Cyndee Tector, Russell Turner, Eli Venter, Aihui H. Wang, Xin Wang, Zhen-Yuan Wang, David A. Wassarman, George M. Weinstock, Jean Weissenbach, Sherita M. Williams, Trevor Woodage, Kim C. Worley, David Wu, Song Yang, Q. Alison Yao, Jane Ye, Ru-Fang Yeh, Jayshree S. Zaveri, Ming Zhan, Guangren Zhang, Qi Zhao, Liansheng Zheng, Xiangqun H. Zheng, Fei N. Zhong, Wenyan Zhong, Xiaojun Zhou, Shiaoping Zhu, Xiaohong Zhu, Hamilton O. Smith, Richard A. Gibbs, Eugene W. Myers, Gerald M. Rubin, and J. Craig Venter. The Genome Sequence of Drosophila melanogaster. *Science*, 287(5461):2185–2195, March 2000.

[61] Udai Bhan Pandey and Charles D. Nichols. Human Disease Models in Drosophila melanogaster and the Role of the Fly in Therapeutic Drug Discovery. *Pharmacological Reviews*, 63(2):411–436, June 2011.

[62] Venkateswara R. Chintapalli, Jing Wang, and Julian A. T. Dow. Using FlyAtlas to identify better *Drosophila melanogaster* models of human disease. *Nature Genetics*, 39(6):715–720, June 2007.

[63] Denise J. Montell, Pernille Rorth, and Allan C. Spradling. slow border cells, a locus required for a developmentally regulated cell migration during oogenesis, encodes Drosophila C EBP. *Cell*, 71(1):51–62, October 1992.

[64] Vladimír Ulman, Martin Maška, Klas E. G. Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu-Yeung Cho, Alexandre C. Dufour, Jean-Christophe Olivo-Marin, Constantino C. Reyes-Aldasoro, Jose A. Solis-Lemus, Robert Bensch, Thomas Brox, Johannes Stegmaier, Ralf Mikut, Steffen Wolf, Fred A. Hamprecht, Tiago Esteves, Pedro Quelhas, Ömer Demirel, Lars Malmström, Florian Jug, Pavel Tomancak, Erik Meijering, Arrate Muñoz-Barrutia, Michal Kozubek, and Carlos Ortiz-de Solorzano. An objective comparison of cell-tracking algorithms. *Nature Methods*, October 2017.

[65] Nikhil R Pal and Sankar K Pal. A review on image segmentation techniques. *Pattern Recognition*, 26(9):1277–1294, September 1993.

[66] E. Meijering. Cell Segmentation: 50 Years Down the Road [Life Sciences]. *IEEE Signal Processing Magazine*, 29(5):140–145, September 2012.

[67] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., February 2006. ISBN 013168728X.

[68] Martin Maška, Vladimír Ulman, David Svoboda, Pavel Matula, Petr Matula, Cristina Ederra, Ainhoa Urbiola, Tomás España, Subramanian Venkatesan, Deepak M. W. Balak, Pavel Karas, Tereza Bolcková, Markéta Streitová, Craig Carthel, Stefano Coraluppi, Nathalie Harder, Karl Rohr, Klas E. G. Magnusson, Joakim Jaldén, Helen M. Blau, Oleh Dzyubachyk, Pavel Křížek, Guy M. Hagen, David Pastor-Escuredo, Daniel Jimenez-Carretero, Maria J. Ledesma-Carbayo, Arrate Muñoz-Barrutia, Erik Meijering, Michal Kozubek, and Carlos Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms. *Bioinformatics (Oxford, England)*, 30(11):1609–1617, June 2014.

[69] Birnie GD. The hl60 cell line: a model system for studying human myeloid cell differentiation. *The British Journal of Cancer. Supplement*, pages 41–5, December 1988.

[70] David Svoboda, Michal Kozubek, and Stanislav Stejskal. Generation of digital phantoms of cell nuclei and simulation of image formation in 3D image cytometry. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 75(6):494–509, June 2009.

[71] José Alonso Solís-Lemus, Brian Stramer, Greg Slabaugh, and Constantino Carlos Reyes-Aldasoro. Segmentation of Overlapping Macrophages Using Anglegram Analysis. In *Medical Image Understanding and Analysis*, Communications in Computer and Information Science, pages 792–803. Springer, Cham, July 2017.

[72] José Alonso Solís-Lemus, Brian Stramer, Greg Slabaugh, and Constantino Carlos Reyes-Aldasoro. Segmentation and Shape Analysis of Macrophages Using Anglegram Analysis. *Journal of Imaging*, 4(1):2, December 2017.

[73] Paul Jaccard. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin Societé Vaudoise des Sciences Naturelles*, 37:547–579, 1901.

[74] Tom Fawcett. An Introduction to ROC Analysis. *Pattern Recognition Letters*, 27(8):861–874, jun Jun 2006.

[75] T. Randen and J. H. Husoy. Filtering for texture classification: a comparative study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4): 291–310, April 1999.

[76] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis and machine vision*, volume 35. Chapman & Hall, January 1994.

[77] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in Computer Vision*, pages 671–679. Elsevier, 1987.

[78] Alan V Oppenheim, Alan S Willsky, and Syed Hamid Nawab. Signals and systems, vol. 2. *Prentice-Hall Englewood Cliffs, NJ*, 6(7):10, 1983.

[79] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November Nov 1986.

[80] Otto H Schmitt. A thermionic trigger. *Journal of Scientific Instruments*, 15(1): 24, 1938.

[81] Nobuyuki Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans Sys, Man, and Cyber*, 9(1):62–66, 1979.

[82] T. N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, 40(4):901–914, April 1992.

[83] Derek Bradley and Gerhard Roth. Adaptive thresholding using the integral image. *Journal of Graphics Tools*, 12(2):13–21, 2007.

[84] C.C. Reyes-Aldasoro. Retrospective shading correction algorithm based on signal envelope estimation. *Electronics Letters*, 45(9):454–456, April 2009.

[85] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[86] I. Fogel and D. Sagi. Gabor Filters As Texture Discriminator. *Biological Cybernetics*, 61(2):103–113, June 1989.

[87] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2, 1999.

[88] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Lecture Notes in Computer Science, pages 404–417. Springer, Berlin, Heidelberg, May 2006.

[89] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, January 1991.

[90] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3): 346–359, June 2008.

[91] Nagham Hamid, Abid Yahya, R Badlishah Ahmad, and Osamah M Al-Qershi. A comparison between using sift and surf for characteristic region based image steganography. *International Journal of Computer Science Issues*, 9(3):110–116, 2012.

[92] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61 (1):38–59, January 1995.

[93] Franz Aurenhammer. Voronoi Diagrams—a Survey of a Fundamental Geometric Data Structure. *ACM Comput. Surv.*, 23(3):345–405, September 1991.

[94] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu, and D. G. Kendall. Algorithms for Computing Voronoi Diagrams. In *Spatial Tessellations*, pages 229–290. John Wiley & Sons, Inc., 2000. ISBN 978-0-470-31701-3.

[95] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, June 2001.

[96] M. E. Plissiti and C. Nikou. Overlapping Cell Nuclei Segmentation Using a Spatially Adaptive Active Physical Model. *IEEE Transactions on Image Processing*, 21(11):4568–4580, November 2012.

[97] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, March 1975.

[98] Guillermo Sapiro. *Geometric partial differential equations and image analysis*. Cambridge university press, 2006.

[99] Z. Lu, G. Carneiro, and A. P. Bradley. An Improved Joint Optimization of Multiple Level Set Functions for the Segmentation of Overlapping Cervical Cells. *IEEE Transactions on Image Processing*, 24(4):1261–1272, April 2015.

[100] Robert T. Collins and Peter Carr. Hybrid Stochastic / Deterministic Optimization for Tracking Sports Players and Pedestrians. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, Lecture Notes in Computer Science, pages 298–313. Springer International Publishing, 2014.

[101] Constantino Carlos Reyes-Aldasoro. Measuring the velocity of fluorescently labelled red blood cells with a keyhole tracking algorithm. *Journal of Microscopy*, 229(1):162–173, 2008.

[102] Katherine M. Henry, Luke Pase, Carlos Fernando Ramos-Lopez, Graham J. Lieschke, Stephen A. Renshaw, and Constantino Carlos Reyes-Aldasoro. PhagoSight: An Open-Source MATLAB® Package for the Analysis of Fluorescent Neutrophil and Macrophage Migration in a Zebrafish Model. *PLOS ONE*, 8(8):e72636, August 2013.

[103] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc 4th Alvey Vision Conf*, page 147–151. Alvety Vision Club, 1988.

[104] T. Lindeberg. Junction detection with automatic selection of detection scales and localization scales. In *Proceedings of 1st International Conference on Image Processing*, volume 1, pages 924–928, Nov 1994.

[105] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1):1–6, November 1998.

[106] Ian Hannah, Devesh Patel, and Roy Davies. The use of variance and entropic thresholding methods for image segmentation. *Pattern Recognition*, 28(8):1135–1143, August Aug 1995.

[107] Myles Hollander, Douglas A. Wolfe, and Eric Chicken. *The One-Sample Location Problem*, pages 39–114. John Wiley & Sons, Inc., 2015.

[108] José Alonso Solís-Lemus, Brian Stramer, Greg Slabaugh, and Csontantino Carlos Reyes-Aldasoro. Shape analysis and tracking of migrating macrophages. pages 1006–1009, April 2018.

[109] José Alonso Solís-Lemus, Brian Stramer, Greg Slabaugh, and Constantino Carlos Reyes-Aldasoro. Analysis of the interactions of migrating macrophages. In Mark Nixon, Sasan Mahmoodi, and Reyer Zwiggelaar, editors, *Medical Image Understanding and Analysis*, pages 262–273, Cham, 2018. Springer International Publishing. ISBN 978-3-319-95921-4.

[110] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

[111] Ross Whitaker. A level-set approach to 3d reconstruction from range data. *Int J Computer Vision*, 29(3):203–231, Sep 1998.

[112] J. Kennedy and R. C. Eberhart. Particle swarm optimization. *IEEE International Conference on Neural Networks*, 4:1942–1948 vol.4, 1995.

[113] José Alonso Solís-Lemus, Brian Stramer, Greg Slabaugh, and Constantino Carlos Reyes-Aldasoro. Macrosight: A novel framework to analyze the shape and movement of interacting macrophages using matlab®. *Journal of Imaging*, 5(1): 17, 2019.

[114] Alice Schoenauer Sebag, Sandra Plancade, Céline Raulet-Tomkiewicz, Robert Barouki, Jean-Philippe Vert, and Thomas Walter. A generic methodological framework for studying single cell motility in high-throughput time-lapse data. *Bioinformatics*, 31(12):i320–i328, June 2015.

[115] Jorge Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York, 2 edition, 2006.

[116] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics, and Image Processing*, 29(3):273–285, March 1985.

[117] R B Fisher, K Dawson-Howe, A Fizgibbon, C Robertson, and E Trucco. *Dictionary of Computer Vision and Image Processing*. Wiley, 2005.

[118] Hai Su, Fujun Liu, Yuanpu Xie, Fuyong Xing, Sreenivasan Meyyappan, and Lin Yang. Region segmentation in histopathological breast cancer images using deep convolutional neural network. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, pages 55–58. IEEE, April 2015. ISBN 978-1-4799-2374-8.

[119] Alessandro Giusti, Dan C. Ciresan, Jonathan Masci, Luca M. Gambardella, and Jurgen Schmidhuber. Fast image scanning with deep max-pooling convolutional neural networks. In *2013 IEEE International Conference on Image Processing*, pages 4034–4038. IEEE, September 2013. ISBN 978-1-4799-2341-0.

[120] Constantino Carlos Reyes-Aldasoro and Ana Laura Aldeco. Image segmentation and compression using neural networks. In *Advances in Artificial Perception and Robotics*, pages 23–25, Guanajuato, Mexico, Oct Oct 2000. CIMAT.

[121] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[122] Vicent Caselles, Ron Kimmel, and Guillermo Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.

Appendix A

Mathematical preliminaries

In this appendix, detailed description of the mathematical techniques is presented. The representation of digital images as matrices allows the utilisation of mathematical techniques involving vector spaces and arrays. The operations performed in images can be linear, nonlinear, they can involve transformations, and changes of domain, for example from spatial to frequency. Because images are considered to be digitalisation of continuous functions, notions in calculus can be applied, the only requirement being taking into consideration the correct numerical application of continuous techniques. This appendix presents the notation used for the operations used throughout the thesis.

A.1 Notation

Throughout this work, vectors will be represented with lowercase bold letters, e.g. $\mathbf{x}, \mathbf{v}, \mathbf{p}$, and matrices will be represented either in uppercase letters, like A, B, C or in blackboard uppercase letters when an image is being referred to, for example $\mathbb{I}, \mathbb{K}$, or a binary image representing a set, for example $\mathbb{A}, \mathbb{B}$. Values in images at a certain location, $\mathbb{I}(x, y)$, are referred to as *pixels*, and are represented by a lowercase p with a lowercase subscript, e.g $p_s = \mathbb{I}(x_s, y_s), p_r = \mathbb{I}(x_r, y_r)$. The plane containing the pixels in the image will be called the *spatial domain*.

A.2 Vector spaces and Matrices

Representing images as matrices allows for all the mathematical developments in vector-matrix theory [67]. In this section, an overview of the elements of linear algebra will be presented, using the classical reference of Nocedal and Wright [115] as a basis

for the more theoretical derivations. Vectors correspond to one-dimensional arrays of numbers which exist in the $\mathbb{R}^n$ space and are generally represented vertically,

$$\mathbf{x} = \begin{pmatrix} a_1 \\ \vdots \\ a_i \\ \vdots \\ a_n \end{pmatrix} = (a_1, \cdots, a_i, \cdots, a_z)^T .$$

Vectors can be operated element by element, as long as they have the same size. An important measurement for vectors is the *interior product*, which multiplies two vectors and produces a scalar,

$$\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b} = \sum_i a_i b_i.$$

As seen before, images will be represented as matrices, in which the space of matrices of size $N_r \times N_c$ can also be represented as $\mathbb{R}^{N_r \times N_c}$. On the other hand, matrices are two dimensional arrays which each entry is represented by two indices, $a_{i,j} = A(i,j)$ where the index $i = 1, \cdots, N_r$ indicates the row number and the index $j = 1, \cdots, N_c$, also, a matrix can be seen as a vertical array of N_r rows, or a horizontal array of N_c columns,

$$A = ((a_{i,j})) = \begin{pmatrix} a_{1,1} & \cdots & a_{1,j} & \cdots & a_{1,N_c} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{i,1} & \cdots & a_{i,j} & \cdots & a_{i,N_c} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ a_{N_r,1} & \cdots & a_{N_r,j} & \cdots & a_{N_r,N_c} \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{a}}_1^T \\ \vdots \\ \hat{\mathbf{a}}_i^T \\ \vdots \\ \hat{\mathbf{a}}_{N_r}^T \end{pmatrix} = (\mathbf{a}_1 \cdots \mathbf{a}_j \cdots \mathbf{a}_{N_r}) .$$

Let $A, B \in \mathbb{R}^{n \times m}$, the operations of addition and scalar product are defined in an element by element basis, *i.e.*,

$$A + B = ((a_{i,j})) + ((b_{i,j})) = ((a_{i,j} + b_{i,j})) ; \quad \alpha A = \alpha \, ((a_{i,j})) = ((\alpha a_{i,j})) .$$

On the other hand, matrix multiplication is defined as

$$AB = \begin{pmatrix} \hat{\mathbf{a}}_1^T \\ \vdots \\ \hat{\mathbf{a}}_i^T \\ \vdots \\ \hat{\mathbf{a}}_{N_r}^T \end{pmatrix} (\mathbf{b}_1 \cdots \mathbf{b}_j \cdots \mathbf{b}_{N_r}) = \left(\left(\hat{\mathbf{a}}_i^T \mathbf{b}_j \right) \right).$$

Finally, array operations are worth mentioning. In which multiplication is done element by element, it is represented by the symbol $*$, for example,

$$A.* B = ((a_{i,j})) \cdot * ((b_{i,j})) = ((a_{i,j} b_{i,j})).$$

Transformations

Transformations modify the spatial relationship between points, in case of curves, and between pixels in the case of images. The principle is very similar, where the only difference lies in the need to interpolate values of pixels after the transformation has occurred [67]. Transformations can be defined in a broad manner, as a function T mapping another $f(x, y)$ into a new one $g(x, y)$, *i.e.*, $g(x, y) = T[f(x, y)]$.

Linear operations

Most transformations presented in this thesis are linear, which imply a modification of scale and translation. More accurately, let $T[\cdot]$ a linear transformation, and f, g functions, then the following properties are fulfilled:

- Let $\alpha \in \mathbb{R}$, then $T[\alpha f] = \alpha T[f]$.
- For functions f, g then $T[f + g] = T[f] + T[g]$.

The previous conditions involve a generalised way of representing linear transformations, however in more practical applications transformations in vector spaces and images occur through matrix operations. The following definition outlines a linear transformation in the context of a vector space.

Definition 7 (Linear transformations) *Let* $\mathbf{x} \in \mathbb{R}^m$ *be a vector and* $A \in \mathbb{R}^{n \times m}$. *Then a linear transformation will be defined as the translation and modification of the*

coordinates of $\mathbf{x}$ *such as*

$$\tilde{\mathbf{x}} = A\mathbf{x} + \mathbf{b}$$

for some matrix $A = (\mathbf{u}_1, \mathbf{u}_2)$ *and vector* $\mathbf{b}$. *Depending on the restrictions of column vectors* $\mathbf{u}$, *is the classification of the transformation.*

Other notions of linear algebra are worth mentioning, like optimisation, linear programming and even some with statistical applications such as least squares and principal component analysis.

Convolution and filtering

As a quick overview of convolution, available in detail in appendix A, let $f(x, y)$ be an image of size $N_r \times N_c$, and let $\mathbb{K}$ be a filter kernel, *i.e.* a small square matrix of size $n^2, n << \min(N_r, N_c)$ with n odd ($n = 2a + 1$). Then the convolution $g(x, y) = f \circledast \mathbb{K}$ is given by Equation (A.1),

$$g(x, y) = \sum_{i=-a}^{a} \sum_{j=-a}^{a} \mathbb{K}(i, j) f(x + i, y + j). \tag{A.1}$$

Notice that convolution can be represented mathematically as in Equation (A.1), with the operator $\circledast$, or in MATLAB® notation, imfilt($\mathbb{I}, \mathbb{K}$).

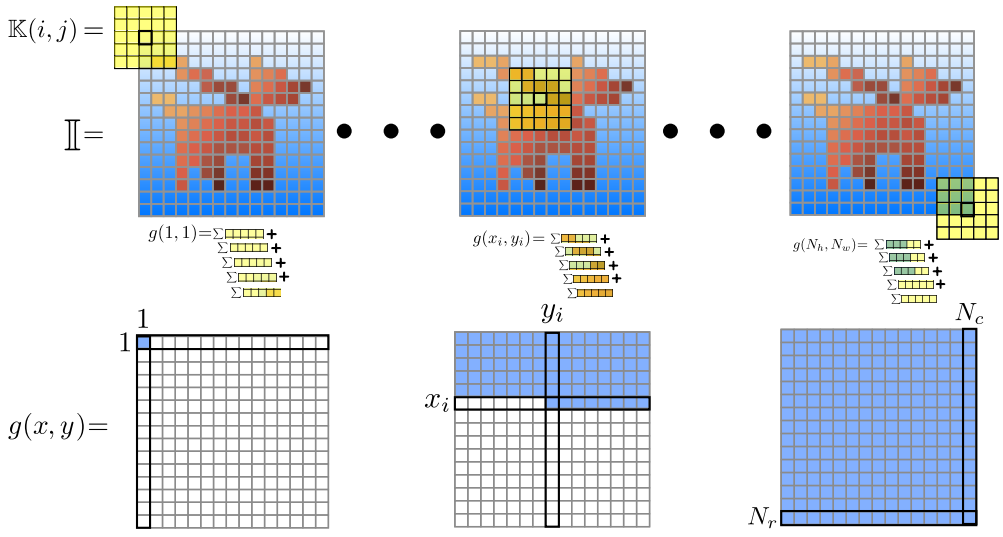


Figure A.1 Detailed diagram of convolution in 2D variables. The diagram shows an image $\mathbb{I}$ and a kernel $\mathbb{K}$ interacting in three different instances; for a particular position (x_i, y_i) in $\mathbb{I}$, a corresponding position in the resulting $g(x, y)$ will be given by the sum of the values of the image, weighted by the values of $\mathbb{K}$.

A.3 Notions of multivariate calculus

Let $f : \mathcal{I} \to \mathbb{R}$, the derivative at point x_0 will be given by the limit of the gradient from points $(x_0 - h, f(x_0 - h))$ to $(x_0 + h, f(x_0 + h))$, like

$$\lim_{h \to 0} \frac{f(x_0 + h) - f(x_0 - h)}{2h}.$$

Regarding discretisation, following the notation from Appendix A, the closest points to discrete variable x are $x - 1$ and $x + 1$. Therefore, the approximation of the derivative will be given by

$$\frac{f(x_0 + 1) - f(x_0 - 1)}{2}, \tag{A.2}$$

the process can be observed in figure A.2(a) and A.2(b). In an image, discretisation of a function with two variables, the process can be extended to include horizontal and vertical directions. Figure A.2(c) shows a representation of the process of estimating the gradient at a point given the two directions x and y.

In multivariate calculus, the individual derivatives with respect to directions x and y are called the partial derivatives $\partial_x f$ and $\partial_y f$. The vector which contains the partial derivatives is called the *gradient*, $\nabla f = (\partial_x f, \partial_y f)^T$. Since the gradient can be calculated per point in the image, two images will be generated containing the partial derivatives per point, highlighting the transitions in intensity at each direction.

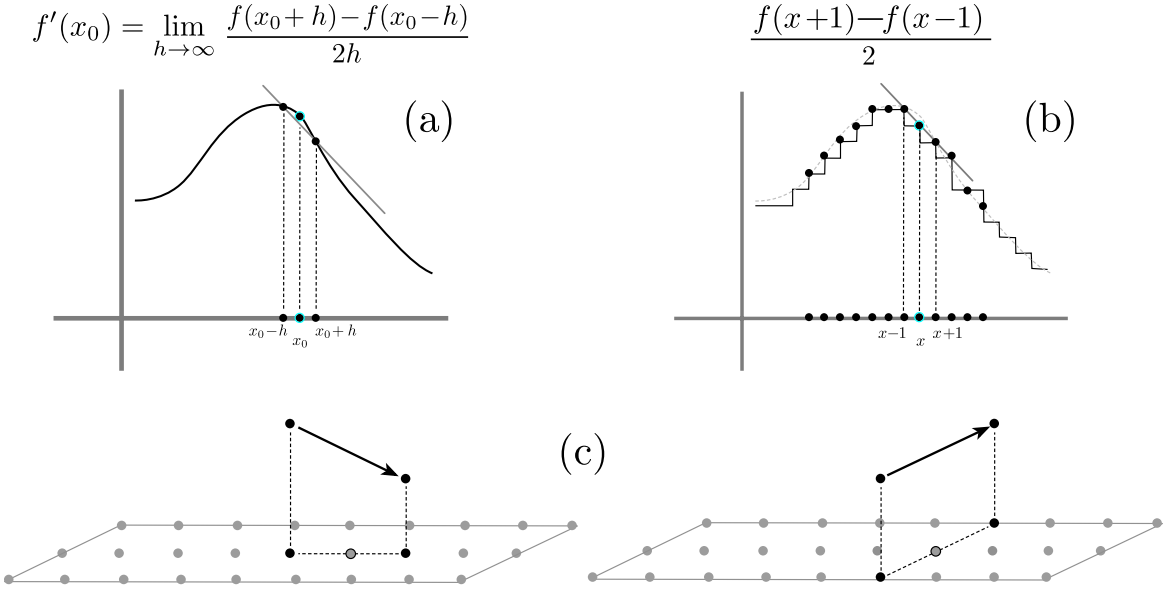


Figure A.2 Discretisation of a central point derivative in 1D variables.

A.4 Set theory

In previous sections, analysis of images has been done analysing them as discretisations of continous functions. objects within images can be described as curves. In this section, the concept of a digital image remains, in which the number of levels is binary, $L = 2$, and normally the levels are integers. Each pixel will take a value, 0 or 1, where contiguous areas of pixels with the same value will form areas and *objects*. In this work, the contiguous areas of pixels with a value of 1 will be considered to be objects or sets. This section will be referred to when exploring the fundamentals of image segmentation and when discussing postprocessing.

In the case of images, the entirety of the ordered pixels within it can be referred to as the *universe* as it contains all possible sets of pixels in it. Thus, each pixel $p = \mathbb{I}(x, y)$ will be an element of the universe. A collection of pixels, will be called a set or a *region* within the image and represented by $\mathbb{A} \subset \mathbb{U}$. Table A.1 shows a brief description of the basic set operations.

Table A.1 Basic set operations. Let $\mathbb{A}, \mathbb{B}$ be sets within the universe $\mathbb{U}$, this table summarises the basic set theory operations adn their definitions.

Name	Symbol	Definition
Union	$\mathbb{A} \cup \mathbb{B}$	$p \in \mathbb{A}$ OR $p \in \mathbb{B}$
Intersection	$\mathbb{A} \cap \mathbb{B}$	$p \in \mathbb{A}$ AND $p \in \mathbb{B}$
Complement	$\mathbb{A}^c$	$p \notin \mathbb{A}$
Difference	$\mathbb{A} - \mathbb{B}$	$p \in \mathbb{A} \cap \mathbb{B}^c$

In the context of images, pixel positions with a particular intensity —commonly 1— will correspond to elements in the sets. Therefore, the regions in an image that constitute a set will be described by positions. In turn, this representation of sets allows for size to be measured in set as the number of elements in the image region.

As mentioned in section 2.1, connected regions in binary images with a value of 1 can be modelled as sets in which the elements correspond to the pixel locations and the universe set corresponds to the entire collection, of pixels, *i.e.* the image $\mathbb{I}$. Let the set $\mathbb{A}$ be a **connected** region in the image $\mathbb{I}$, then the set will be defined as the collection of points (x, y) in where $\mathbb{A} = \{(x, y) | \mathbb{I}(x, y) = 1\}$. Therefore, cardinality of a set $|\mathbb{A}|$, also referred to as size of the set, is defined by the number of connected pixels in a given region. Basic set operations are defined in Table A.2.

Table A.2 Basic set operations. Let $\mathbb{A}, \mathbb{B}$ be sets within the universe $\mathbb{U}$ and let $p \in \mathbb{U}$ be an element in the universe. This table summarises the basic set theory operations and their definitions.

Name	Symbol	Definition
Union	$\mathbb{A} \cup \mathbb{B}$	$p \in \mathbb{A}$ OR $p \in \mathbb{B}$
Intersection	$\mathbb{A} \cap \mathbb{B}$	$p \in \mathbb{A}$ AND $p \in \mathbb{B}$
Complement	$\mathbb{A}^c$	$p \notin \mathbb{A}$
Difference	$\mathbb{A} - \mathbb{B}$	$p \in \mathbb{A} \cap \mathbb{B}^c$

To avoid confusing notation, only this section, uses the notation of sets to present the operations in abstract; however, in practical terms, the sets will be referred to as regions and computations, like the ones described in section 3.1, will be described in terms of binary images.

Probability and statistics. In image analysis, probability and statistics appear in numerous occasions, from simple descriptive statistics of the distributions of the intensities, to modelling of the shapes and boundaries of objects in the image. This section is focused on the descriptive operations and graphics in images.

To illustrate basic concepts in statistics, let $\{p_x\}_{x=1}^n$ be the set of pixel intensities in an image $\mathbb{I} \in \mathbb{R}^{N_r, N_c}$, then the **mean**, μ_I and **variance** σ_I^2 are defined by equation (A.3),

$$\mu_I = \sum_x \frac{p_x}{N_r N_c}, \quad \sigma_I^2 = \sum_x \frac{(p_x - \mu_I)^2}{N_r N_c}. \tag{A.3}$$

Considering sets of images, $\{\mathbb{I}_t\}_1^T$, then the previous definitions can be extended, and the mean and variable frames can be computed. Consider the images to be aligned in a 3D array $\mathbb{I}^{3D}$ of size $N_r \times N_c \times T$, then in equation (A.4), the extensions of the statistics are outlined,

$$\mu_I^{3D} = \frac{1}{T} \sum_t \mathbb{I}_t, \quad (\sigma_I^{3D})^2 = \frac{1}{T} \sum_t (\mathbb{I}_t - \mu_I^{3D})^2. \tag{A.4}$$

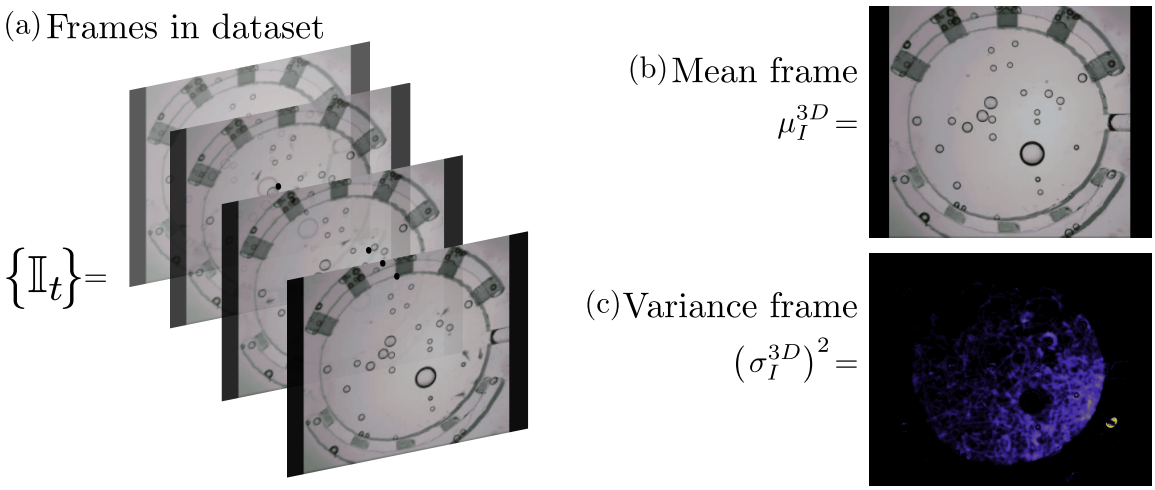


Figure A.3 Representation of the mean and variance frames in the Artemia Franciscana dataset. (a) Represents the collection of frames $\mathbb{I}_t$ of which the mean and variance are calculated. The operations for mean and variance are performed per pixel. Notice the mean frame (b) in the Artemia Franciscana dataset displays parts of the image frames without movement, while the variance frame (c) represents the areas of the image where movement was involved.

Statistic descriptive graphs are also recurrent in analysis of datasets. Two classical graphs that aid in the visualisation of the observed distribution are the **histogram** and **box-plots**. Such representations can be intuitive to some extent to describe the distribution variable. In sections 2.2 and 2.2, histograms are displayed describing the distribution of the pixels present in each presented frame. In formal terms, a histogram is a graphic in which the values of the variable, such as the pixel intensities, is categorised into ranges of values called *bins*. On the other hand, a **box-plot** is a graphic which shows the quartiles of the variable, the minimum, maximum values and the outliers.

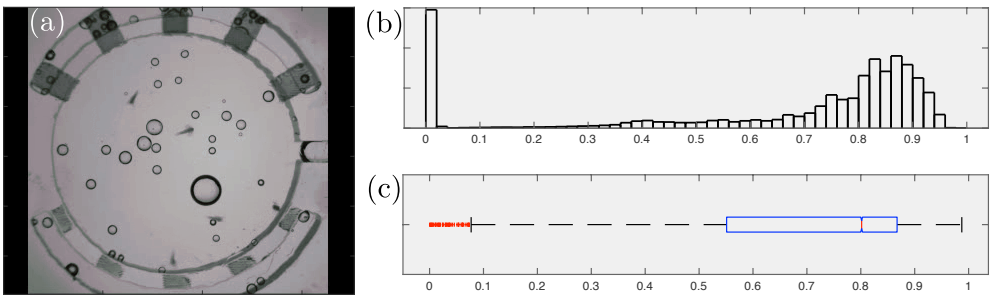


Figure A.4 Representation of a histogram of the intensity values of an image. (a) Shows a frame in the *Artemia franciscana* dataset. (b) Shows the histogram of the intensity values in the image and (c) its boxplot graph

Appendix B

Preliminaries of image analysis

In this appendix, an overview on the broad field of image analysis is presented. Classical approaches are overviewed and in some cases tested on generic images. The techniques discussed here present a thorough explanation of the methods used as exploration of the data in chapter 3.

B.1 Classical image analysis

In this work, three main stages in image processing can be recognised, which will be analysed in the following sections. The stages are preprocessing, which refer to the intensity based operations performed on an image to aid in the segmentation techniques perform better. Then, the core process of image segmentation will be discussed; such process involves the classification of each pixel into one of two categories: background and foreground. Finally, a postprocessing stage follows in which small errors in segmentation are addressed, this stage usually involves some type of binary operation. Sections 3.1, 3.1 and 3.1 show the relevant techniques in each category, applied to the data presented in sections 2.2, 1.8 and 2.2.

Preprocessing

Preprocessing of an image in this report refers to the operations performed on an image that allows for a more precise segmentation or analysis of the data. These operations can highlight some of the images' features, remove noise, modify dimensions and should improve the results of posterior operations. One classical classification of preprocessing techniques is from the work by Sonka et al. [76], which classifies

such techniques into pixel brightness transformations, geometric transformations and transformations in local neighbourhoods. Pixel brightness transformations refer to modifications in the image regarding the pixels themselves, which normally address brightness or grey-scale corrections. Geometric transformations are made to address different types of distortion an image may present. Finally, local-preprocessing involves the operation in the neighbourhood of a given pixel in order to produce a new value for that location.

Furthermore, another approach in the work by Gonzalez and Woods [67] classifies the techniques into transformations of intensities and spatial filtering. Transformations of intensities, refer to transformations to the intensity levels $I = 1, 2 \cdots, L$ in each pixel, which produce modifications of the statistical distribution of the intensities. Spatial filtering refers to the local neighbourhood operations discussed in [76]. Techniques which involve the convolution of the image with a filter kernel, to produce a new image in which each pixel is the result of a local operation. In this work, the preprocessing techniques used fall into the local-preprocessing and pixel brightness categories. Figure B.1 shows a graphical overview of the techniques analysed in this work and the classification it has in classical literature.

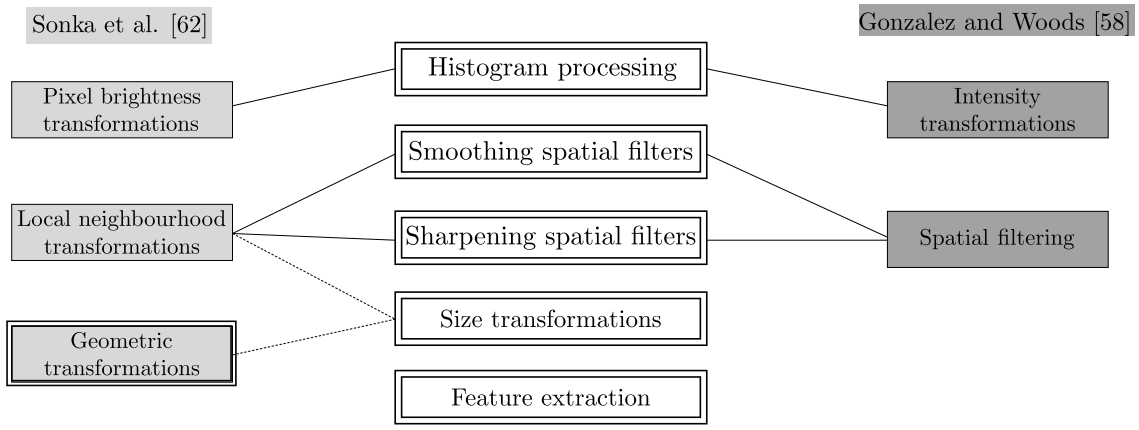


Figure B.1 Overview of the preprocessing methodologies analysed in this work. In the centre of the diagram (double line) all the techniques as studied in the upcoming sections.

Histogram modifications

Every pixel of the image will change its own value based in a determined in a function. The transformation $T[\mathbb{I}(x, y)]$ can be rewritten as a one-dimensional function applied to each pixel $T(p)$, it is worth noting that the location is not relevant to this transformation,

only the intensity level at each point. Some examples of histogram modification are described in Table B.1.

Table B.1 Histogram modification functions. Let p_r be a pixel in an image, which can be one of the intensity levels $I = 1, 2, \ldots, L$; and let p_s be the modified value after applying the function $T(\cdot)$, $p_s = T(p_r)$.

Function on pixel	Description
$p_s = L - p_r$	Negatives, used to invert the levels
$p_s = c \log(1 + p_r)$	Logarithms
$p_s = cp_r^\gamma$	Power-law or Gamma corrections
Contrast stretching	Changing the range of pixel intensity values.
Histogram equalisation	Increase the global contrast of an image.

Size transformations

The objective of this section is to observe different features in the image which at each resolution, it can also aid by increasing the signal to noise ratio of the image and decreasing the computational complexity as this would reduce the number of computations performed on the image. Two alternatives are overviewed: first, a more simplified reduction in size of an image by interpolation and then a Gaussian pyramid, which includes a more sophisticated way of using the values in the original image to create the lower resolution ones. Further implementations of scaling and operations at different scales involve sort of filters and scale space analysis [104], both fall out of the scope of this work.

Reduction in size and Interpolation A common transformation is to reduce the size of the image in order to increase the signal to noise ratio [77]. The reduction can be as straightforward as taking windows of size $m \times n$ pixels and, just like in spatial filtering techniques, calculating some statistic of the values inside the window, for example the mean, the maximum or minimum.

Reduction with a Gaussian pyramid. In their work, Burt and Adelson [77] developed an algorithm to reduce an image. Given an input $\mathbb{I}_0 \in \mathbb{R}^{m \times n}$, then the image will be reduced to an output $\mathbb{I}_1$ of size $\lceil m/2 \rceil \times \lceil n/2 \rceil$, which will be a reduced, low-pass filter version of $\mathbb{I}_0$. The way the reduction is performed is through averaging each collection of 5×5 pixels with the weights found by a matrix in which each entry is given by

$W(q,r) = \hat{w}(q)\hat{w}(r), \quad r,q = 1,2,\cdots,5$, where $\hat{w} = (1/4 - a/2, 1/4, a, 1/4, 1/4 - a/2)$, figure B.3.

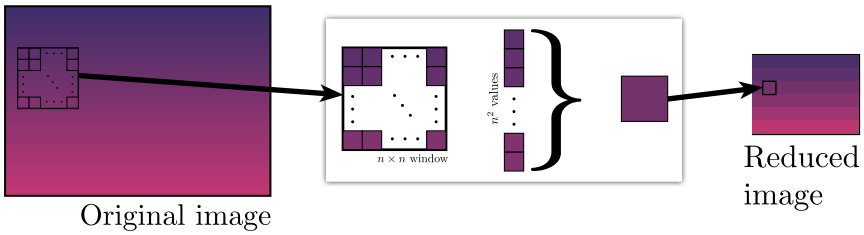
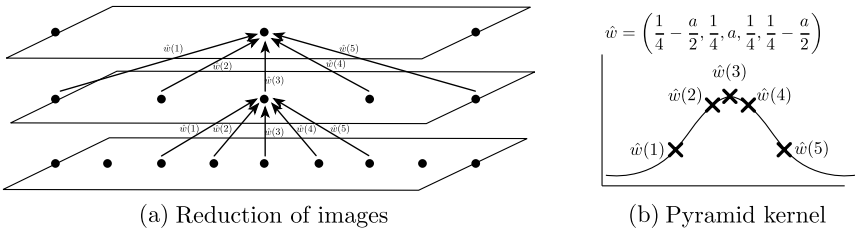


Figure B.2 Illustration of size reduction in an image. The image is represented in the left with a vertical change in colour and an $n \times n$ window moving through it. The centre image shows the $n \times n$ window and its n^2 values which are summarised by some function like mean or maximum. The output of the procedure is an image reduced in size, with reduced resolution.



(a) Reduction of images (b) Pyramid kernel

Figure B.3 Representation of the process which generates a Gaussian pyramid in one dimension. (a) Displays the reduction process, where dots represent locations (pixels) in the image, each of which is weighed by the values of pyramid kernel (b). In it, the value of a can be changed to alter desired effects; in the case of a Gaussian reduction, a specific value of $a \approx 3$ is used to simulate the shape of a Gaussian.

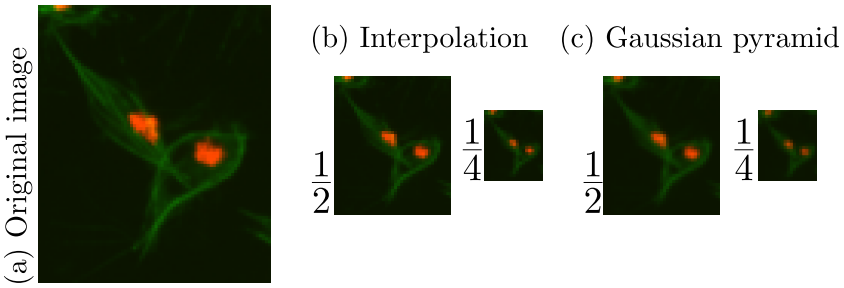


Figure B.4 Examples of size reduction using both things proposed. From the original image (a), size reduction is performed twice, to create an image of half the size, and an image of 25% the size. (b) Shows the outputs for the image when using interpolation and (c) shows the images when using a Gaussian pyramid.

Geometric transformations

In this section, geometric transformations are described as modifications of the spatial relationship between the pixels in an image. Linear transformations are described by a matrix $\mathbb{T} \in \mathbb{R}^{3\times3}$. From definition 7 in appendix A.2, three types of linear transformations can be identified:

Affine transformations. There are no restrictions in matrix $\mathbb{T}$. Such transformations can cause the loss of information such as the area of the object inside the boundary.

Euclidean transformations. When $\mathbb{T} = (\mathbf{u}_1, \mathbf{u}_2)$ is such that $\langle \mathbf{u}_1, \mathbf{u}_2 \rangle = 0$ and $|\mathbf{u}| = 1$.

Equi-affine transformations. When $\det(\mathbb{T}) = 1$, then the area of the curve transformed will be preserved.

Geometric transformations are shown for completeness, although they are not a part the scope of this work. Transformations can be defined via matrices which alter the positions (x, y) in an image, allowing it to change its size, the ratios of their rows and columns and translating it to a different point. The transformation matrix $\mathbb{T}$ is of size 3×3. Table B.2 represents the different matrices used for transformations.

Table B.2 Geometric transformations represented in 3×3 matrices. The linear transformations are represented by the matrices $\mathbb{T}$.

	Matrix $\mathbb{T}$	Description
Translation	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{pmatrix}$	(t_x, t_y) are the displacements along the axes.
Scale	$\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$	(s_x, s_y) specify the scales along the the axes.
Rotation	$\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Rotation by angle θ.
Shear	$\begin{pmatrix} 1 & h_x & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	(h_x, h_y) specify the shear factor along the axes.

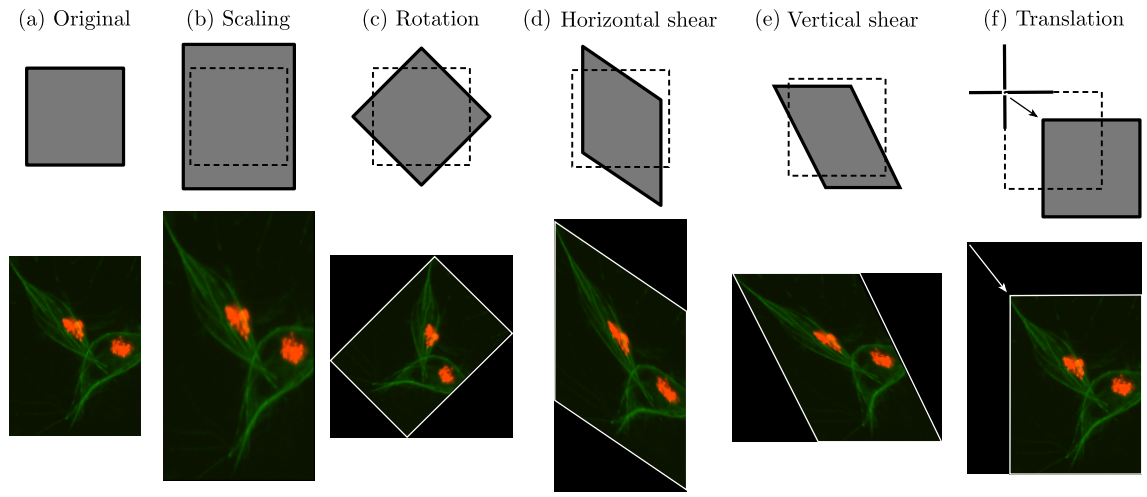


Figure B.5 Examples of transformations on detail of a frame. The top row shows the transformations with an abstract 2D object, with the dotted lines representing the original shape. White lines have been added for visualisation purposes.

Filtering

This section provides an overview of the techniques in which each pixel in the image is assigned a specific neighbourhood in which a predefined operation is performed, [67, Chapter 3]. Filtering can be viewed differently depending on the field of study. Apart from reducing noise in the images, other filters can be used to enhance some properties or features of the image, extracting relevant information from it. The term *filtering* is taken from the field of digital signal analysis and processing in which properties of a signal are selected, enhanced or suppressed through convolution with a finite function with special characteristics. For a thorough review of convolution in 2D digital signals, the reader is referred to section A, while for Signal Analysis and Processing, the classical works by Oppenheim et al. [78] and [67] are recommended.

Filter kernels consist of matrices commonly of small sizes that can have different parameters that determine the values within them. Throughout this work, it can be assumed that all kernels, represented with letter $\mathbb{K}$, are square matrices of size n, with $n = 2a + 1, a \in \mathbb{N}$, unless explicitly determined.

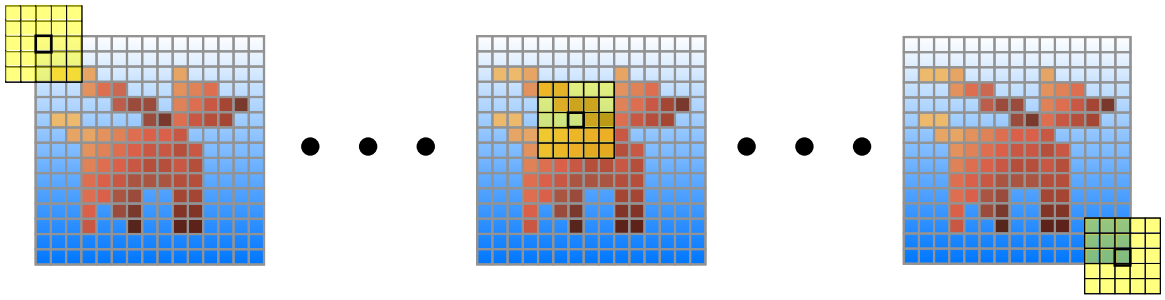


Figure B.6 Simplified diagram of a 2D convolution of an image and a 5×5 kernel. This is a simplified version of the diagram in figure A.1. In this case, the kernel, $\mathbb{K}$, is represented by a 5×5 matrix (yellow) as it moves onto different parts of the image. The position of the origin in $\mathbb{K}$ will be the position in the resulting image.

Depending on the type of filter applied, the values will change to highlight certain features in the image, like edges or corners. The following sections discuss two of the principal filtering approaches: smoothing and sharpening of image features. The difference lies in the properties of the kernels chosen to perform the convolution.

Smoothing filters

In this section, examples of smoothing filters are presented, as well as its application on relevant images for this work. The filters are presented in terms of the kernels that contain them.

Averaging kernel In this type of kernels, the values of the elements in $\mathbb{K}$ represents a weight or proportion, that is, the values are between $(0, 1)$ and the sum of all the values in the kernel add up to 1. With the previous explanation in mind, in the convolution $\mathbb{I} \circledast \mathbb{K} = \text{imfilt}(\mathbb{I}, \mathbb{K})$, every pixel value is replaced by the weighted average of the pixels in the n-neighbourhood.

The most straightforward implementation of this filter is what is known as a **box filter**, which has uniform weights assigned to all the pixels $\mathbb{K}(i, j) = 1/n^2$. The only parameter is the size of the filter, which at each point in the resulting matrix produces the mean of the intensity values contained within the scope of the filter. Another example of the weight distribution in the kernel is the **disk filter**, an averaging filter acting upon a circular area. The circle is or radius r and the size of the kernel is $2r + 1$. The filter highlights circular structures which match the size of the radius of the circle. The principle is the same as the one in the box filter, given the size of the filter, taking the central element $\mathbb{K}(r + 1, r + 1)$, all elements within the radius will contain a value

higher than zero. As it is a discrete filter, the edges of the circle will not coincide with the finite grid of the kernel.

Gaussian filtering The Gaussian filter is another type of smoothing filter in which the values follow a discrete 2D Gaussian of the size of the kernel. The parameters in the filter are zero mean (0) and variance (σ^2), as well as the size of the filter (n). As seen in section A, the 2D shape of the Gaussian resembles a bell, where the width of it is determined by the variance. It is important to note that the Gaussian kernel must consider a value for n large enough for an adequate discretisation of $\mathbb{K}$.

Order statistic filtering. Most filtering is performed through convolution. As shown in figure B.6, the convolution can be interpreted as a moving *window* of the size of the kernel, which at every step, selects the pixels in the input image that will contribute to the new intensities certain position in the output; normally through the sum of the array multiplication of the elements in the kernel by the elements in the image.

Order statistic filtering, the same idea of a moving window is taken, but at every stage, the calculation of a percentile is performed in the pixels selected at a certain moment. The most common calculations are the minimum, maximum and median filters. Figure B.8 displays the detail of three examples of order statistic filtering: minimum, median and maximum filtering with various different sizes. Finally, figure B.7 displays different cases in greater detail.

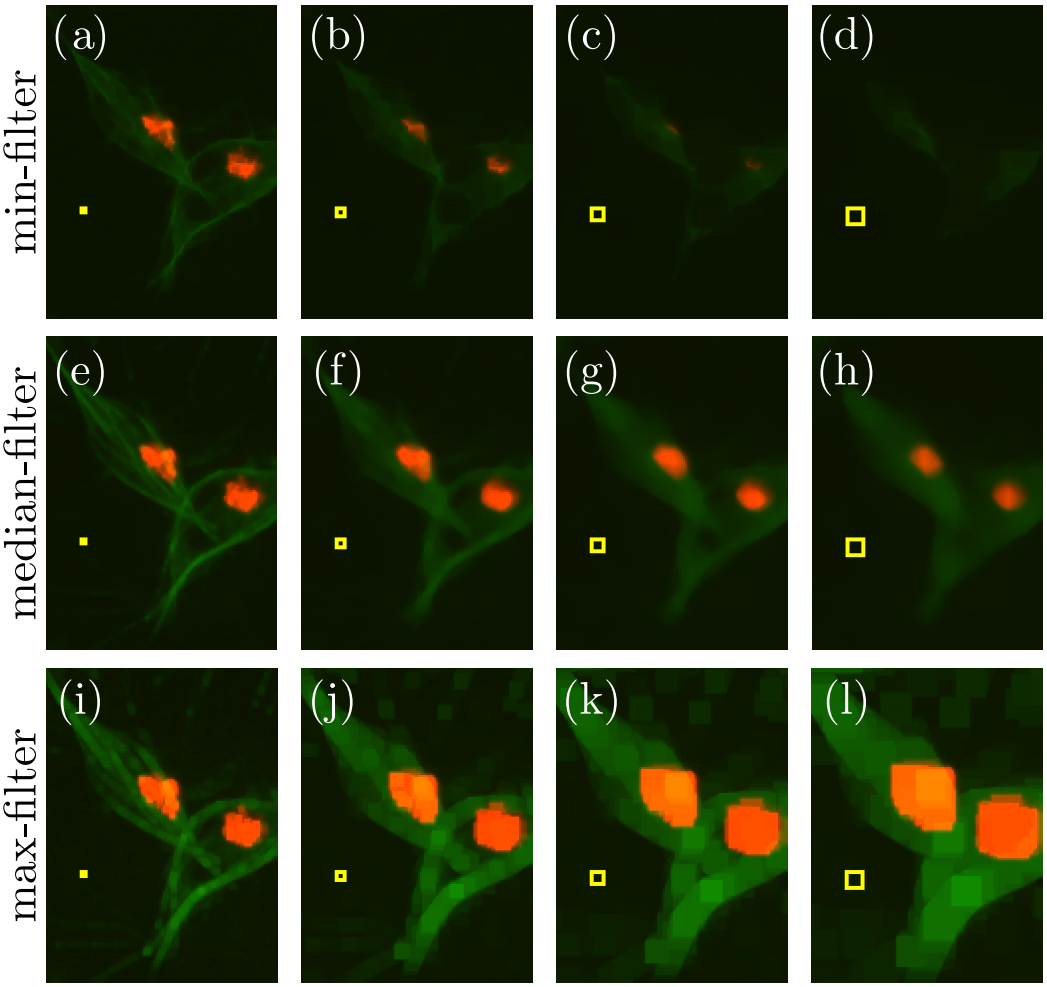
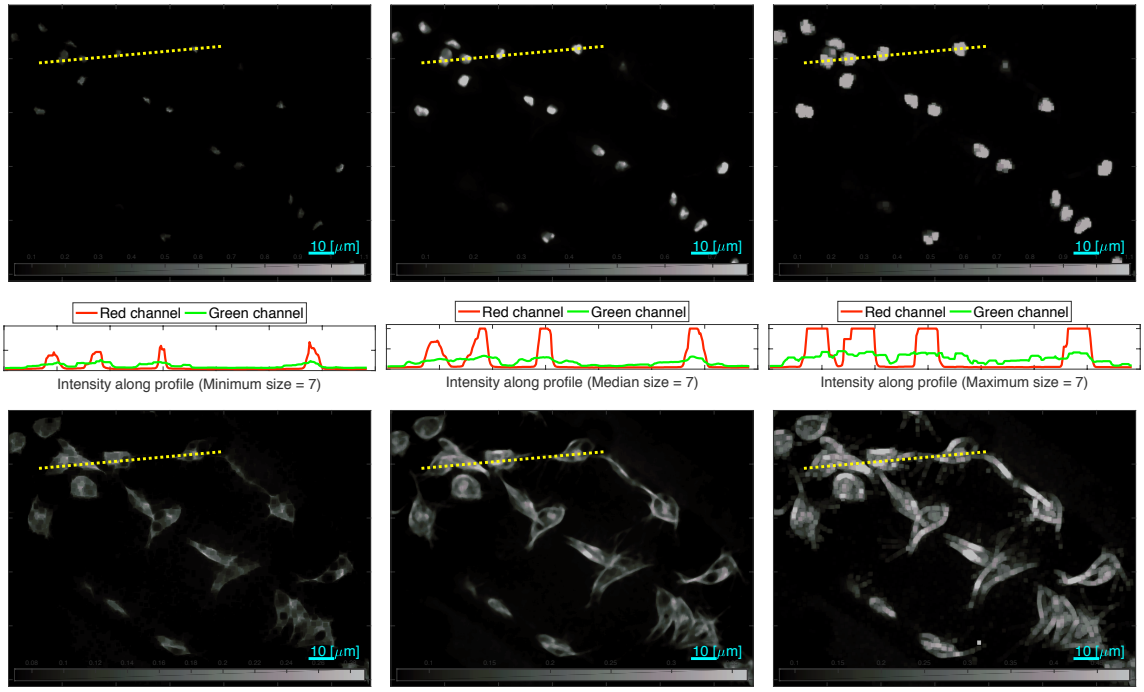


Figure B.7 Representation of order different statistic filters with varying sizes. Detail of a frame is presented after applying three types of order statistic filters: minimum (top row: (a-d)), median (middle row: (e-h)) and maximum (bottom row: (i-l)). Each filter was tried using different sizes, and applied to each of the two channels to showcase the behaviour of each technique. A yellow box was superimposed to the image to aid the visualisation of the size of the filter. From the left: first column (a,e,i) 3×3; second column (b,f,j) 7×7; third column (c,g,k) 11×11; fourth column (d,h,l) 15×15

In general, smoothing filters are useful if the data presents high frequency noise, and thus serve as *low pass filters*, in the context of signal analysis. However, some features in the image are also of high frequency, like abrupt changes in the intensities, which could be due to the presence of an edge or points in the image. For that purpose, sharpening filters are described in the following sections.



(a) Minimum filter. (b) Median filter (c) Maximum filter

Figure B.8 Test frame after applying smoothing through different Order statistic filters.

Sharpening filters

To analyse a digital image, it is sometimes useful to locate certain geometric objects, such as lines which could correspond to edges and ridges in the image where certain key objects are located. Thus, highlighting transitions in intensity, with a sense of the magnitude of such transitions becomes key to image processing [67].

In calculus, the operator that describes the rate of change at a certain point is the derivative, or in multivariate calculus, the gradient. As images were constructed as discretisations of continuous functions of two variables, applying the numerical concepts of differentiation to the images would provide an insight to the characterisation of intensity transitions. On the other hand, as an image can be observed as a signal in the spatial domain, abrupt transitions could be linked to locations of a high frequency. In figure B.9, a combination of both ideas is represented, as this section describes the tools used to highlight transitions in images.

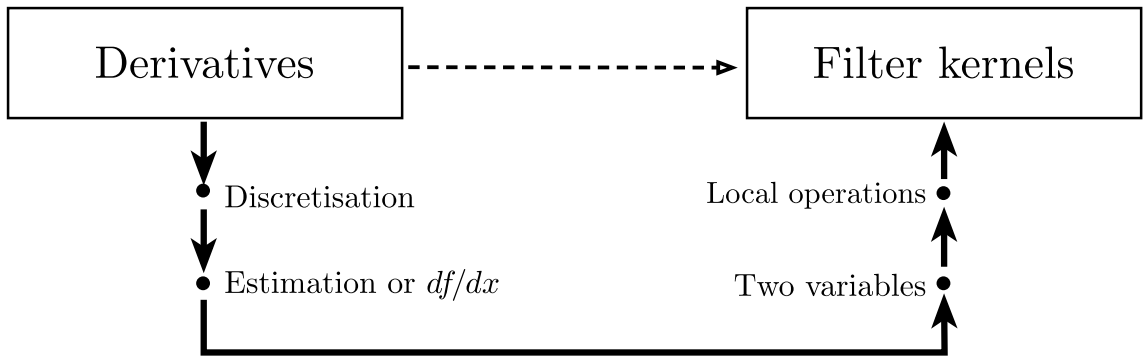


Figure B.9 Relationship between derivatives and filters. In this section, the relationship between a derivative approximation for a 2D discrete function and a filter approach in which local operations performed for each point —or pixel— in the function.

The notion of differentiation comes from the analysis of continuous variables in calculus, therefore, a numerical interpretation of the derivative in a discrete space must be made to adapt to the discrete images. The notion then needs to be extended into multiple variables, to account for the two directions present in images, rows and columns. Finally, the notion of local will emerge, which will in turn provide the background to present an approach which can be linked to spatial filtering.

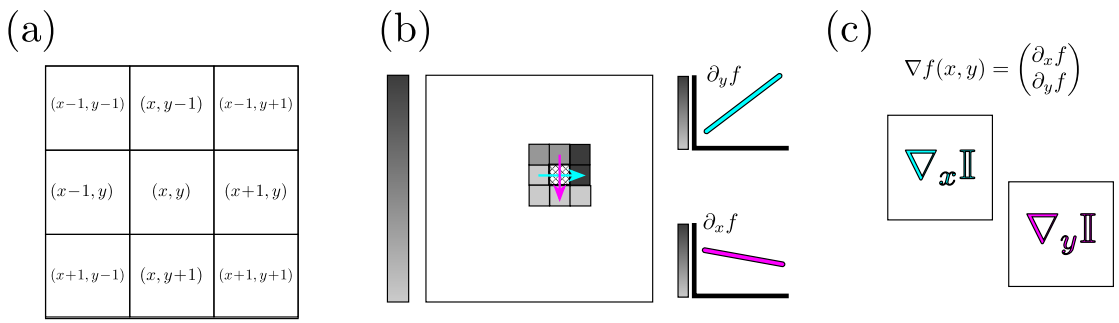


Figure B.10 Explanation of the image gradient, recalling figure A.2. Given a pixel at a position (x, y), the neighbourhood close to it can be determined by the pixels at positions $x-1, x+1$ and $y-1, y+1$, such a neighbourhood is shown in (a). A representation of the two different partial derivatives can be seen in (b), where a pixel can show changes in both directions. (c) Shows a representation of the gradient in the direction x, $\nabla_x \mathbb{I}$ and in direction y, $\nabla_y \mathbb{I}$.

Methods based on derivatives

In this section, methods based on derivatives will be described. The derivatives will be outlined both in terms of the gradient and the filter which can be used to perform the analysis.

Central differences and image gradient. This section will build on the principles used for upcoming filter implementations. The central differences approximation to the derivative has already been presented in equation (A.2). To extend it to two variables, the procedure is calculated separately in both directions, at each position (x, y),

$$\partial_x \mathbb{I}(x,y) = \frac{\mathbb{I}(x+1,y) - \mathbb{I}(x-1,y)}{2}, \tag{B.1}$$

$$\partial_y \mathbb{I}(x,y) = \frac{\mathbb{I}(x,y+1) - \mathbb{I}(x,y-1)}{2}. \tag{B.2}$$

In this work, the symbol used for partial derivatives applied to an image,

$$\partial_x \mathbb{I}(x,y), \partial_y \mathbb{I}(x,y),$$

will refer to the pointwise estimation of the derivative. While the symbol for gradient $\nabla_x \mathbb{I}$ will refer to the image which contains all the partial derivative calculations per position, *i.e.* $\nabla_x \mathbb{I} = (\partial_x \mathbb{I}(i,j))$. Consider a kernels given by the matrices $\mathbb{K}_x = (1,0,1)^T$ and $\mathbb{K}_y = (1,0,1)$. Centred around position (x,y), a convolution of certain filters in the image would produce at each pixel the approximation of the derivative per point. The image gradient will consist of both images containing all the partial derivatives per point, and per direction: $\mathbb{G}_x = \nabla_x \mathbb{I}, \mathbb{G}_y = \nabla_y \mathbb{I}$. It is common to produce the image gradient magnitude, calculating per element $\mathbb{G} = |\nabla \mathbb{I}| = \sqrt{\nabla_x \mathbb{I}^2 + \nabla_y \mathbb{I}^2}$.

In the following paragraphs, different experimentations of the idea derived here will be presented.

Sobel gradient. In the Sobel gradient, the corresponding kernels are given by equation (B.3):

$$\mathbb{K}_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \mathbb{K}_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{B.3}$$

Using the previous kernels, it can be observed that the value estimated for $\partial_x \mathbb{I}(x, y)$ will be given by equation (B.4).

$$\begin{aligned} \partial_x \mathbb{I}(x, y) \quad = \quad & \mathbb{I}(x+1, y-1) - \mathbb{I}(x-1, y-1) \\ & +2\left(\mathbb{I}(x+1, y) - \mathbb{I}(x-1, y)\right) \\ & +\mathbb{I}(x+1, y+1) - \mathbb{I}(x-1, y+1) \end{aligned} \tag{B.4}$$

This will give the pixel $\partial_x \mathbb{I}(x, y)$ a sum of the estimations of the derivatives at $(x, y-1)$ and $(x, y+1)$. The equation for $\partial_x \mathbb{I}(x, y)$ would be analogue to the one presented in equation (B.4).

Prewitt gradient. The process with the Prewitt gradient is similar to that of Sobel's. The difference comes with the kernel, which are given by equation (B.5),

$$\mathbb{K}_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \mathbb{K}_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \tag{B.5}$$

comparing the Prewitt kernel to Sobel's, note the central row/column point has a higher value assigned to it.

Roberts gradient. Finally, the Roberts' gradient presents a variant, however following the same idea. A cross gradient operator is introduced to highlight diagonal transitions in intensities. The kernels used are given by equation (B.6),

$$\mathbb{K}_{d1} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad \mathbb{K}_{d2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}. \tag{B.6}$$

To showcase the different implementations of the derivatives, figure B.11 includes the detail of a frame in the `MACROS1` dataset in which each channel has been processed with each of the filters. In all cases, the image gradient $\mathbb{G} = |\nabla \mathbb{I}| = \sqrt{\nabla_x \mathbb{I}^2 + \nabla_y \mathbb{I}^2}$ is shown.

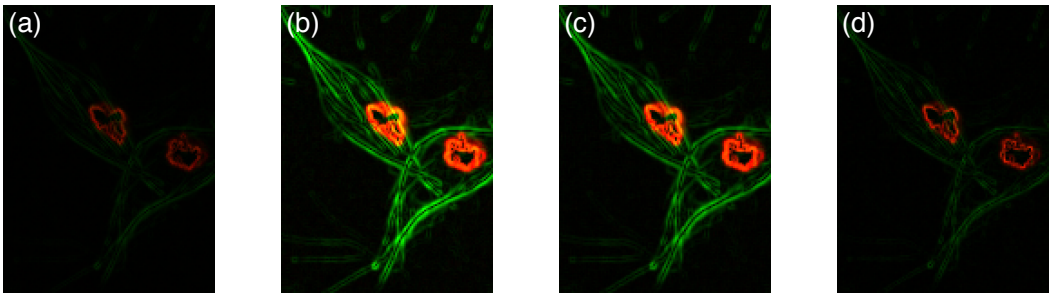
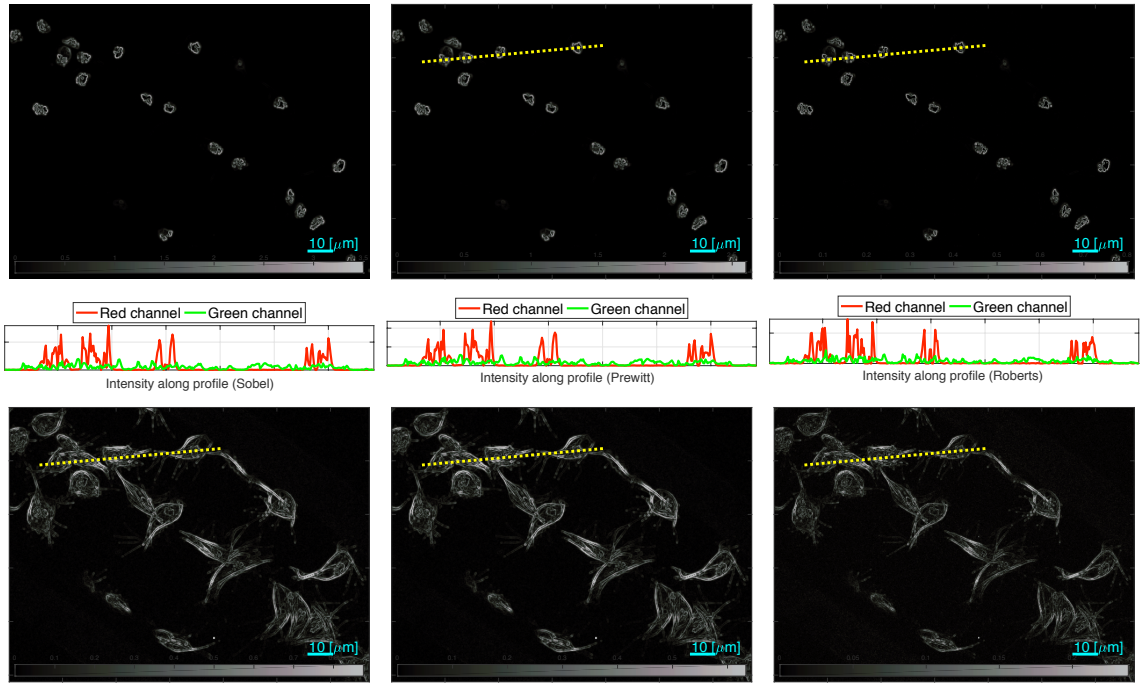


Figure B.11 Representation of different sharpening filters. Detail of a frame is presented after applying the different sharpening filters analysed in this section, each one presenting the gradient's magnitude $\mathbb{G} = \sqrt{\mathbb{G}_x^2 + \mathbb{G}_y^2}$. (a) Central differences, (b) Sobel, (c) Prewitt and (d) Roberts.



(a) Sobel gradient. (b) Prewitt gradient. (c) Roberts gradient.

Figure B.12 Test frame after applying smoothing through different filters.

Laplacian filter. The construction of the Laplacian filter is done from the definition of the Laplacian operator and the intermediate difference estimation of the gradient —as opposed to the central differences used before. The Laplacian operator is given by equation (B.7),

$$\nabla^2 f = \partial_{xx}^2 f + \partial_{yy}^2 f. \tag{B.7}$$

The first order intermediate difference estimator with respect to x is $\partial_x f = f(x+1, y) - f(x, y)$. Thus the estimator for the second order partial derivative with respect to x will be given by

$$\begin{aligned} \partial^2_{xx} f &= \partial_x f(x+1, y) - \partial_x f(x, y) \\ &= f\left((x+1)+1, y\right) - f(x+1, y) - [f(x+1, y) - f(x, y)] \\ &= f(x+2, y) + f(x, y) - 2f(x+1, y) \end{aligned}$$

Evaluating the previous equation in $x - 1$ yields the estimator

$$\partial^2_{xx} f = f(x+1, y) + f(x-1, y) - 2f(x, y),$$

and thus, using the analogous estimator for $\partial^2_y f$ into equation (B.7), yields the following equation (B.8):

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y). \tag{B.8}$$

Notice that the values included in the filter will correspond to the values within the 8 neighbours of position (x, y), which can be visualised in figure B.10(b). The kernels formed will be given by equation (B.9),

$$\mathbb{K}_1 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \mathbb{K}_2 = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \tag{B.9}$$

where $\mathbb{K}_1$ is the one described in this section, and $\mathbb{K}_2$ is an extension which includes the cross sectional elements. To highlight the gradients differently, the arithmetic inverse of each kernel can be used. Figure B.13 shows an image after applying both kernels and their negatives.

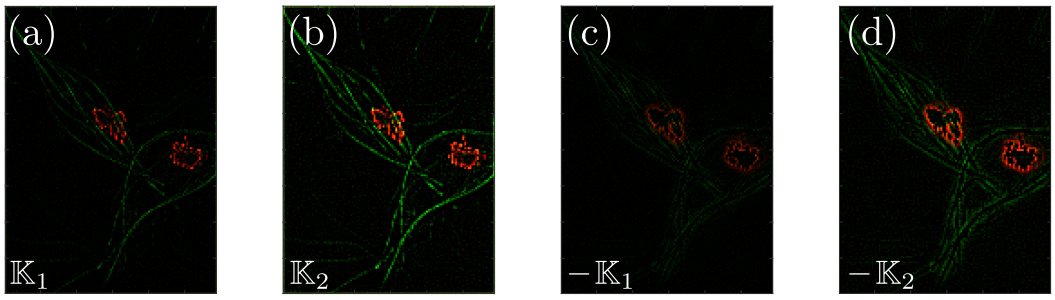


Figure B.13 Representation of the different implementations of the Laplacian filter for $\mathbb{K}_1$ and $\mathbb{K}_2$. (a) $\mathbb{K}_1$, (b) $\mathbb{K}_2$, (c) $-\mathbb{K}_1$ and (d) $-\mathbb{K}_2$.

Applications of the Hessian matrix. Other implementations of second order derivatives include the Hessian matrix, which is a 2×2 matrix containing the second order derivatives with respect to x, y and the cross derivatives:

$$\mathbb{H}(f) = \begin{bmatrix} \partial^2_{xx} & \partial^2_{xy} \\ \partial^2_{yx} & \partial^2_{yy} \end{bmatrix} = \begin{bmatrix} \mathbb{G}_{xx} & \mathbb{G}_{xy} \\ \mathbb{G}_{xy} & \mathbb{G}_{yy} \end{bmatrix}.$$

As this matrix is produced per point, then four matrices will be calculated per image. This procedure is useful when some operations of the Hessian values are needed. For example, blob detection, done by the determinant of the Hessian $\text{DoH} = \mathbb{G}_{xx} \cdot * \mathbb{G}_{yy} - \mathbb{G}_{xy} \cdot \wedge 2$, or in **ridge detection**, where the eigenvalues of the Hessian per pixel are found by

$$\lambda_{\max}(\mathbb{H}) = \mathbb{G}_{xx} + \mathbb{G}_{yy} \pm \sqrt{\mathbb{G}_{xx} \cdot \wedge 2 + \mathbb{G}_{yy} \cdot \wedge 2 - 2\mathbb{G}_{xx}\mathbb{G}_{yy} + 4\mathbb{G}_{xy} \cdot \wedge 2}.$$

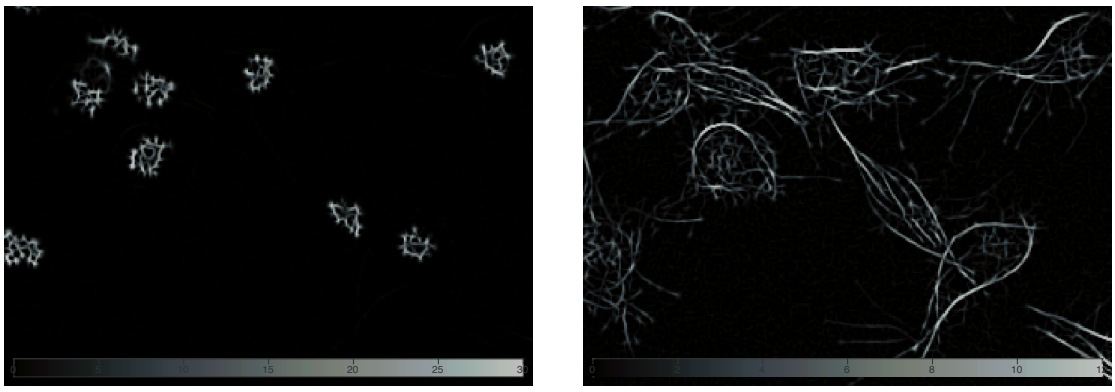


Figure B.14 Example of the minimum eigenvalue of the Hessian matrix per pixel.

Edge detection An important problem in image analysis is the detection of contours of objects or edges. Note that filtering highlights edges and points of abrupt changes, however edge detection refers to the creation of a binary image where positive values are located where changes are more abrupt. In particular, the Canny [79] edge detection algorithm has been a widely referenced image analysis technique with almost 30 thousand citations since its publication in 1986[1]. As it is relevant to this work, an overview of the algorithm is provided and some examples of its implementations, while varying its key parameters. The overview of the algorithm is described in Algorithm 3. Let $\mathbb{I}$ be an image, then its edges are found by looking for local maxima in its gradient, $\mathbb{G} = |\nabla \mathbb{I}|$. The function used to highlight the transitions is the derivative of a Gaussian filter with zero-mean and a variance σ^2 provided by the user. The Canny [79] performs a hysteresis threshold, which involves the definition of two thresholds, weak and strong, where all intensities above the strong threshold are kept, and intensities between the thresholds are only kept if they are in contact with a strong edge. The process of hysteresis is overviewed in section 3.1.

Algorithm 3: Canny algorithm

Input: Image: $\mathbb{I}$, Standard deviation of Gaussian: σ
Output: Binary image with selected edges: $\mathbb{B}_{\mathbb{I},\sigma}$

Let $\mathbb{K}_\sigma \sim \mathcal{N}(0, \sigma^2)$ of size $a \times a$, $a = 2n + 1$;
$\mathbb{I}_\mathbb{K} = \mathbb{I} \circledast \mathbb{K}_\sigma$ % Gaussian filter to smooth the image
$[\mathbb{G}_x, \mathbb{G}_y] = \text{imgradient}(\mathbb{I}_\mathbb{K})$ % Compute gradients of the image in each direction (x, y)
$\mathbb{G} = \sqrt{\mathbb{G}_x \cdot \wedge 2 + \mathbb{G}_y \cdot \wedge 2}$ % Compute magnitude of gradient
$\bar{\mathbb{G}}$ = Non-maximum suppression of $\mathbb{G}$
$\mathbb{B}_{\mathbb{I},\sigma}$ = Hysteresis thresholding of $\bar{\mathbb{G}}$ using thresholds $k_1^\star, k_2^\star$.

The two main advantages of the algorithm involves the ability to modify the Gaussian kernels, which allows different edges to be visible and the non-maximum suppression which allows to select the more relevant ones. The technique would be better classified as a segmentation of the edges in the image, as the output is a binary image with two levels, $I = 0, 1$. It is included in this section to provide context of the application of sharpening filters and because the term Segmentation, in this work, refers to the distinction of cells from other cells and from the background.

[1] Consulted in Google Scholar, August 2018

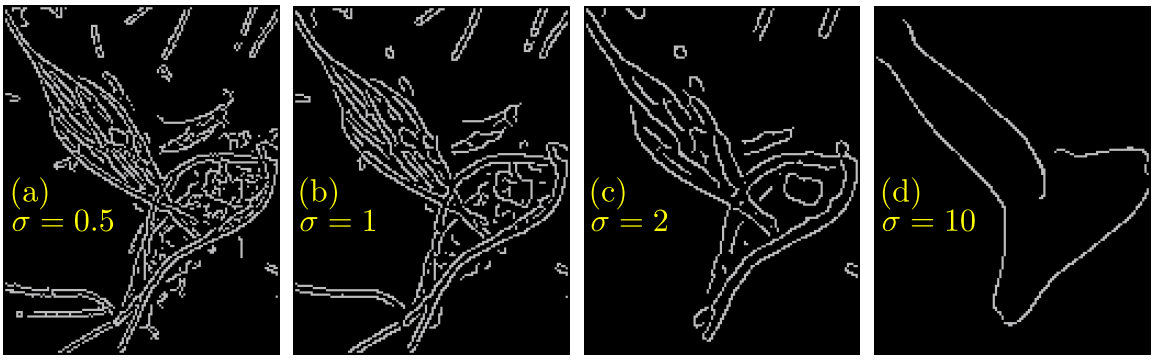


Figure B.15 Detail of output of Canny algorithm on macrophages data. Different values of σ^2 are presented to showcase the output of the algorithm. Notice that the size of the variance σ is inversely proportional to the level of detail being analysed by the algorithm.

Image segmentation

As mentioned in previous sections, segmentation is the process of classifying pixels into one of two categories: foreground, or objects of interest, and background. In a way, segmentation provides context to an image selecting the regions which —given an application— are important. In particular, cell segmentation is a widely studied problem which has produced considerable amount of research output, like the works by Maška et al. [68] and Ulman et al. [64].

Segmentation by intensity thresholding is central in this work, the technique is explored. However, due to its simplicity, it introduces ambiguity in the segmentation when used with the macrophages datasets, due to the interacting cells, which cause overlapping. Thus, more sophisticated approaches are explored to resolve the ambiguity. Apart from intensity thresholding, and to solve segmentation in overlapping cases, more complex techniques are explored. Starting these with a naïve approach based in Voronoi Tesselations, and then moving on to curve evolution techniques such as active contours and level set methods. Then, methods categorised as machine learning are described, ranging from Active Shape Models (ASM), Self-Organising Maps (SOM) and Convolutional Neural Networks (CNN).

Intensity Thresholding

Intensity thresholding is a segmentation technique in which the intensity levels of a signal, or image, are divided into groups allocated based on a set of one or more levels against which the intensities are compared. In the simplest case of one threshold, elements of the signal that are below or above the threshold are considered as background

and the opposite are considered as the foreground or objects of interest. This is a simple and effective method, since, when considering greyscale images, one can imagine that an object should be different in intensity from the background, assuming that the feature that discriminates them is the grey level intensity. In cell analysis, the underlying assumption is that cells have a consistent and significant difference in intensity than the background. The algorithms that are studied in this report separate the images into classes, with grey level intervals corresponding to different classes.

Considering the definitions in section 2.1, intensity levels on an image are referred to be a finite set of L levels, $\{1, 2, 3, \ldots, L\}$. Thus, performing a segmentation by intensity thresholding implies finding a value k which separates the intensity levels into groups: $\{1, 2, 3, \ldots, k-1\}$ and $\{k, k+1, k+2, \ldots, L\}$. Let $\mathcal{C}_R$ be the set of intensities in which the intensities of an image are categorised, where R refers to the region in which the image will be segmented, and can be either a subscript or a number depending on the context. It is important to remember that the segmentation output of a method will be a binary image which takes the value 1 at the detections of foreground and 0 at the background. As it is relevant to this work, the concept of hysteresis thresholding —mentioned in section 3.1— is explained below.

Hysteresis thresholding Hysteresis thresholding is a technique for image segmentation that uses two levels to segment an image, a higher one that determines what will be part of the foreground and a lower one that will distinguish what will be considered as background. The original idea was proposed by Schmitt [80] and has been extended to a two (or three) dimensional space. Regions that are below the lower threshold and are connected to the higher threshold will be retained; while the regions below the lower thresholds surrounded by background will be discarded.

Formally, let $I = \{1, 2, 3, \ldots, L\}$ the levels of intensity in an image $\mathbb{I}$ and let $k_0, k_1 \in I$, such that $k_0 < k_1$. Then, the interval gets partitioned into three classes: $\mathcal{C}_0 = \{1, 2, 3, \ldots, k_0 - 1\}$, $\mathcal{C}_1 = \{k_0, k_0 + 1, k_0 + 2, \ldots, k_1 - 1\}$ and $\mathcal{C}_2 = \{k_1, k_1 + 1, \ldots L\}$. Given the intensity $p_x = \mathbb{I}(x, y)$, three possibilities can be observed for the corresponding pixel in the output image $b = \mathbb{B}(x, y)$.

1. $p < k_0 - 1$, then $p \in \mathcal{C}_0$, and $b = 0$.

2. $p \geq k_1$, then $p \in \mathcal{C}_2$, and $b = 1$.

3. $k_0 \leq p < k2$, then $p \in \mathcal{C}_1$, then:

(a) $b = 1$ if the region surrounding p is connected to a pixel in $\mathcal{C}_2$, or

(b) $b = 0$ if p is surrounded by pixels in region $\mathcal{C}_0$.

Figure B.16 shows a diagram of hysteresis performed in a 1D signal. The same concept can be applied in 2D trivially, to showcase this, figures B.17 and B.18 show an example using a frame of the green channel in a macrophages dataset in different regions of interest of different sizes.

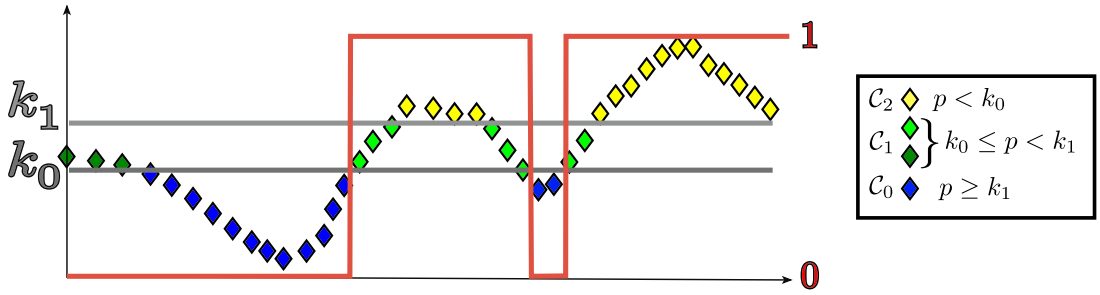


Figure B.16 Diagram of hysteresis in one dimension. The threshold levels $k_0 < k_1$ are shown in shades of grey. Diamond markers ($\diamond$) represent the signal and the different colours represent their relationship to the thresholds; blue colour represents values belonging to $\mathcal{C}_0$ and are assigned 0, green shades represent points in $\mathcal{C}_1$, with a lighter shade representing the markers that will be assigned 1. Finally, yellow markers show the points which belong in $\mathcal{C}_2$ and are automatically assigned 1. The output binary signal is displayed in red.

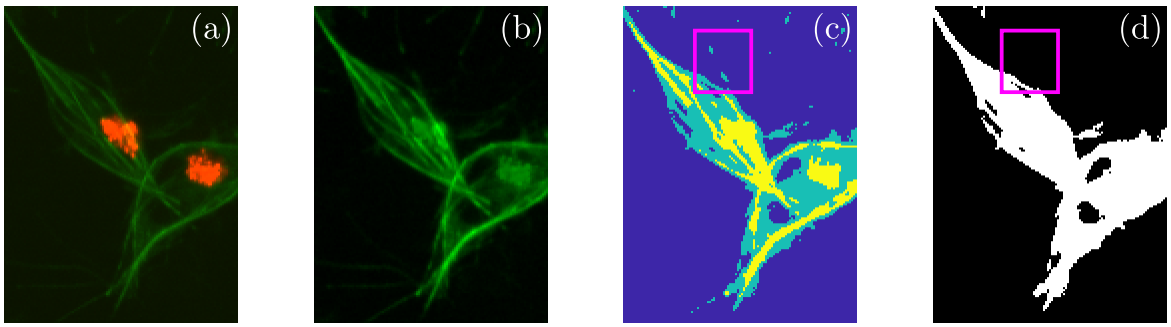


Figure B.17 Detail of hysteresis threshold in macrophages frame. From the RGB macrophage frame (a), the green channel is selected (b). Two thresholds are selected and the resulting example is shown in (c), three classes are thus created. Positions where the levels of intensities fall within a certain class are marked in a different colour, blue for $\mathcal{C}_0$, green for $\mathcal{C}_1$ and yellow for $\mathcal{C}_2$. The result of the hysteresis process appears in (d).

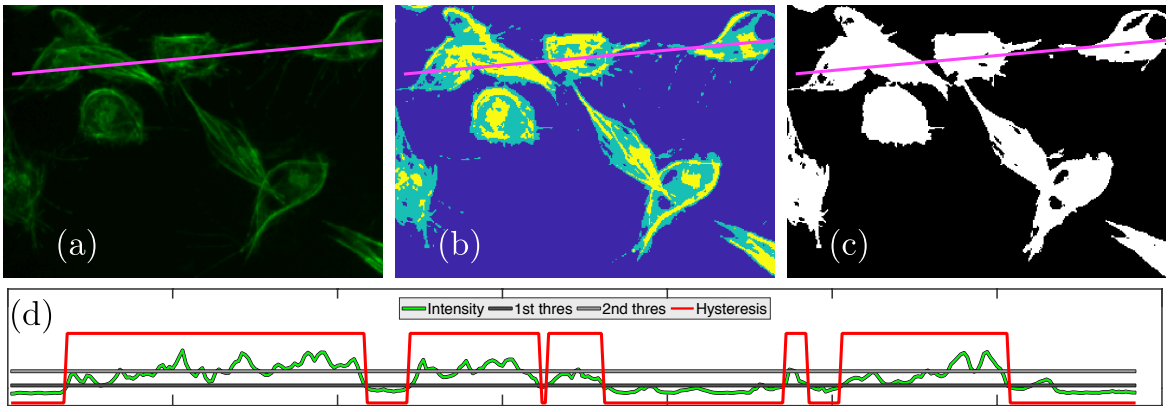


Figure B.18 Illustration of a hysteresis threshold in a 2D image.

Otsu's threshold

Otsu [81] developed an algorithm for differentiating between classes, *i.e.* finding a threshold, by maximising the quotient of *between* and *within* variances of the classes. In simple terms, the algorithm would select a threshold that would distinguish the classes as much as possible, while at the same time would make the intensity levels within the class as alike as possible, one advantage is that this procedure can simply be extended to more than two classes. The derivation of the algorithm is presented for a single threshold as the extension to multiple thresholds is straightforward.

Let $N = n_1, n_2, \cdots, n_L$ be the number pixels of an image $\mathbb{I}$, *i.e.* $n_L = N_r \times N_c$, then each n_i corresponds to the number of pixels of intensity $i \in I$ where I is a finite set of grey level intensities, normally $I = \{1, 2, 3, \cdots, L\}$. Thus, the probability of a pixel $q_x = \mathbb{I}(x, y)$ in the image to belong to each level can be computed as $p_i = P\{x = i\} = n_i/N$. As the goal of this method is to determine the two disjoint classes $\mathcal{C}_0, \mathcal{C}_1$, separated by a threshold $1 \leq k \leq L$, then the probability of a pixel q_x belonging to each class is computed by

$$P\{x \in \mathcal{C}_0\} = \omega_0 = \sum_i^k p_i, \quad \text{and} \tag{B.10a}$$

$$P\{x \in \mathcal{C}_1\} = \omega_1 = 1 - \omega_0 \tag{B.10b}$$

The main objective of the method consists in finding $k^\star$ that maximises the quotient σ_B^2/σ_W^2, where:

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2, \tag{B.11a}$$

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2. \tag{B.11b}$$

The pairs of parameters (μ_j, σ_j^2), $j = \{0, 1\}$ correspond to the conditional mean and variance of the intensity level. The total mean and variance of the intensities in I is then $\mu_T = \omega_0\mu_0 + \omega_1\mu_1$. The work by Otsu [81] can be generalised in a straightforward manner to two or more thresholds by expanding all the definitions in equations (B.10) and (B.11) to having different thresholds. In this work, two thresholds will be determined to perform then a hysteresis threshold, then the classes would be extended in terms of two thresholds k_0, k_1, as $1 \leq k_0 < \ldots < k_1 \leq L$, simply by redefining the classes as $\mathcal{C}_0 = \{1, \ldots, k_0\}, \mathcal{C}_1 = \{k_0 + 1, \ldots, k_1\}$ and $\mathcal{C}_2 = \{k_1 + 1, \ldots, L\}$.

The adaptation of the equations is simple enough, and in the end, the quotient that must be maximised remains the same, and the terms from equations (B.11) change into those in equation (B.12):

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 + \omega_2(\mu_2 - \mu_T)^2, \tag{B.12a}$$

$$\sigma_W^2 = \omega_0\sigma_0^2 + \omega_1\sigma_1^2 + \omega_2\sigma_2^2, \tag{B.12b}$$

the process previously described can easily be rewritten for more thresholds that, depending on the application, could lead to better segmentation, or a better visualisation of the data. Figure B.19 displays the detail of a macrophages frame and the regions created after intensities get classified after applying different Otsu thresholds. Then,

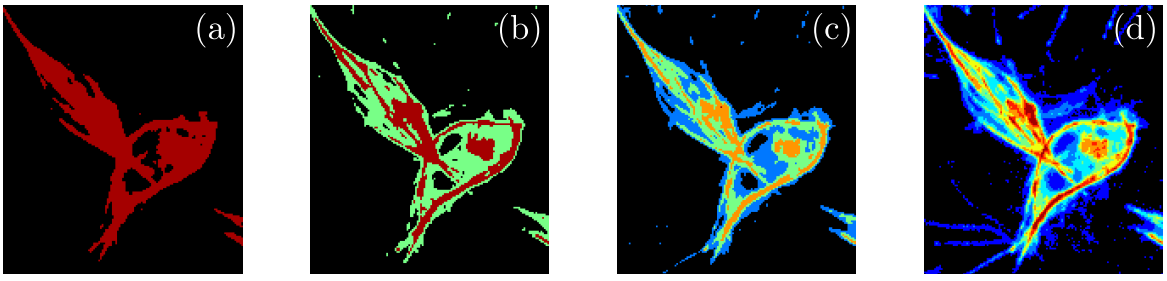


Figure B.19 Detail of two macrophages after applying different Otsu thresholds. The Otsu algorithm was applied to the same section of a frame. Different number of thresholds were applied, producing different number of classes $\mathcal{C}_i$. (a-d) show the results of applying 1,2, 4 and 8 thresholds selected by the Otsu method.

Otsu's method, while very fast, is very sensitive to outliers on the data. One example can be studied on the ISBI datasets, with the Fluo-N2DH-GOWT1 dataset. It can be observed in figure B.20(b) how the difference between the most and least intense is too high for the algorithm to segment both cells. Also, it can be noted in B.20(b) that even taking a robust approach like a hysteresis threshold is not enough for cells of this kind.

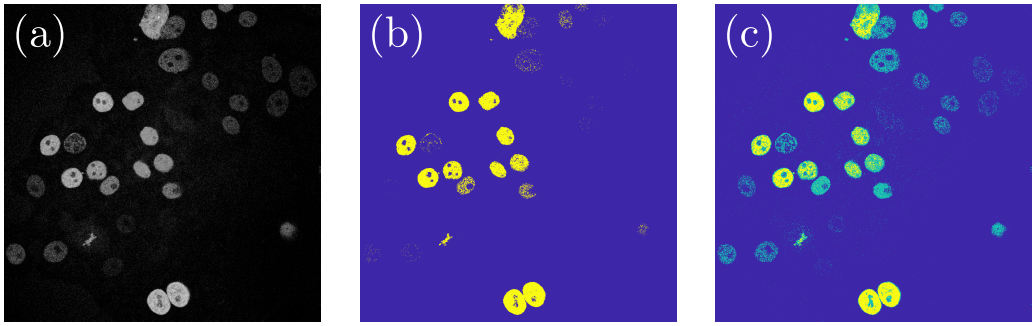


Figure B.20 Limitations of the Otsu algorithm in frame from the ISBI dataset. (a) Represents a frame from the Fluo-N2DH-GOWT1 dataset, recalling section 2.2. One and two thresholds were applied to the image, and the resulting classifications of the pixels are presented in (b) and (c), respectively. It is important to notice that the variability of intensities in the foreground is too large, and the algorithm is incapable of segmenting some of the darker cells.

Entropy based thresholding

Entropy-based thresholding [106] is a methodology in which a threshold is selected based on optimising an objective function, like in the past section, but changing the function to be optimised. The algorithm, originally described by Kapur et al. [116], has the same premise as Otsu's method, but instead of selecting the threshold that maximises the quotient of the between and within variances, the objective function becomes Shannon's entropy. Using the same notation as before, the entropy of a class $\mathcal{C}$ can be defined by (B.13):

$$H\left(\mathcal{C}\right) = -\sum_{i \in C} q_i \log\left(q_i\right), \tag{B.13}$$

where $\mathcal{C}$ is a class, and $q_i = p_i / \sum_{i \in \mathcal{C}} p_i$. The algorithm exploits the notion that there is a relationship between the entropy of an image region with the information it can hold. Therefore, the difference in the classes selected would differ from the classes selected by Otsu's algorithm. The mechanism of maximising a function that depends

on the threshold selected remains, but the driving force of the method is significantly different.

Adaptive thresholding

Variations of the thresholding technique is presented in the works by Pappas [82] and Bradley and Roth [83]. In this work, a thorough derivation of the work by Pappas [82] is presented, however a brief summary of the work by Bradley and Roth [83], along with results on the data are presented for comparison.

In [82], the segmentation achieved was different from the traditional ones, as the threshold level chosen is dependant on the local pixels at given positions. Instead of selecting a set of thresholds that take into consideration the statistical properties of the entire image, the method starts with global estimates of the levels and slowly adapts them to the local characteristics of each region.

The concept of an adaptive threshold becomes relevant to this problem, as the objects of interest in fluorescent images can vary considerably, and a single threshold might not be enough, even a robust one like a hysteresis threshold, as seen in figure B.20. In addition, it is possible that the images suffer from shading due to a variety of reasons [84]. The work by Pappas [82] can be seen as a generalisation of the k-means algorithm [85] that include spatial constraints and account for local variations of intensity. The algorithm works through the estimation of the regions of an image (segmentation) and the parameters in an *a posteriori* density function that computes the probability of the distribution of regions (x) given the observed image (y) as shown in equation (B.14)

$$p\left(x|y\right) \propto p\left(y|x\right) p(x), \tag{B.14}$$

where $y|x \sim \mathcal{N}(\mu, \sigma^2)$ and $p(x|y)$ is the conditional density of the observed image given the distribution of regions and $p(x)$ is the *a priori* density of the region process. Equation (B.15) shows both $p\left(y|x\right)$ and $p(x)$:

$$p(y|x) \propto \exp\left\{-\sum_s \frac{1}{2\sigma^2}\left(y_s - \mu_s^{x_s}\right)^2\right\}, \tag{B.15a}$$

$$p(x) \propto \exp\left\{-\sum_c V_c(x)\right\}, \tag{B.15b}$$

where the subscript s corresponds the location of a pixel on the image, x_s corresponds to the region assigned to pixel s, so $\mu_s^{x_s}$ is the mean value of the image region x_s centred

at s. Finally, the *a priori* density of the region process is modelled with a Gibbs density that sums over the *cliques* C of the region, which compares the similarities of the neighbouring pixels, thus introducing accountability on local regions and restrictions. Figure B.21 shows the process of computing $\mu_s^{x_s}$ where the image is divided into regions and a window scans the image and computes the mean intensities of the selected area. It is important to understand that the superscript in $\mu_s^{x_s}$ is not related to the power operation, but as an index of the different regions that have been selected on the image. The region process density is the element that gives the spatial constraints to the algorithm, and it is modelled as a Gibbs Markov random field.

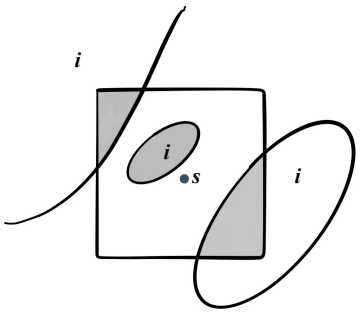


Figure B.21 Graphical explanation of the calculation of parameter $\mu_s^{x_s}$. For the example, the region belonging to pixel s is defined by number i, so $x_s = i$. To compute $\mu_s^{x_s}$, one selects a window inside the image and takes the mean from the areas that belong to that particular region $x_s = i$.

On the other hand, the work by Bradley and Roth [83] presents an adaptive threshold methodology that takes into account spatial variations in illumination. It achieves its purpose by selecting a threshold value based on local mean intensity in the neighbourhood of each pixel. A neighbourhood of 1/8th of the size of the image is used around each pixel. Each pixel gets assigned a different threshold value in which each pixel is compared against. A parameter called *sensitivity* is included, which allows for some control from the user into the threshold taken by the algorithm.

Postprocessing

Postprocessing is the set of operations performed on binary images which aid the segmentation to reduce noise or over-detection. For example, in the case of thresholding, some of the high frequency noise within the image could cause some noise to be detected. However, such detected regions would not have the same shape properties as true detections. Therefore, operations in binary images become relevant to *filtering* out

some of the noise detected by the segmentation; in other applications, the segmentation requirements could be of analysis of simpler shapes, so some simple morphological techniques could be used to modify the detected objects.

In this section, some techniques of morphology are overviewed. Morphology in this work refers to *mathematical morphology*, the branch of mathematics that deals with the representation and description of region shapes in a space [67]. As mentioned before, the images are assumed to be binary, *i.e.* the number of intensity levels is two, and thus represented by the set $I = \{0, 1\}$. The positions of each pixels are still given by (x, y) where $x = 1, 2, \cdots, N_r$ and $y = 1, 2, \cdots, N_c$. In the context of segmentation of cells, some of the operations can aid in reducing noise, smoothing the edges of the shapes and help give a more robust segmentation. Measurements can be performed on the binary images, like area and orientation.

In section 3.1, the construction of filter kernels and the notion of convolution was explored in order to modify the intensities by assigning an operation performed on a neighbourhood of a pixel. To extend the notion into binary images and sets, some operations must be defined in the context of sets, and then translated into binary arrays to define each of the operations.

Structuring elements and notation

In this work, sets are represented by connected regions in an image with the same value. The equivalent of a kernel, will be a small binary matrix which containing a region which would highlight some properties in regions of a binary image. As mentioned in the classic work [67, Chapter 10], a *structuring element* is a small set of subimage used to probe an image under study for properties of interest. Such objects can be thought of in abstract as sets with a centre, or reference point, or as a binary kernel, represented by the symbol $\mathbb{S}$.

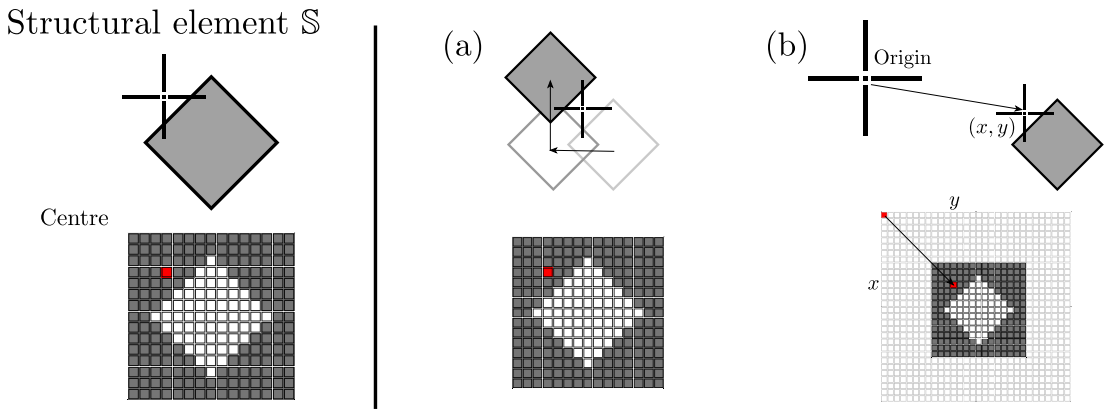


Figure B.22 Representation of a structural element in abstract terms and as a binary kernel $\mathbb{S}$. The diagram shows the abstract representation of a structuring element as a set with a particular shape. The notion of a centre refers to a point of reference in which the element can pivot and from which it can move in a plane. (a) Shows the operation of reflection while (b) shows translation. Notice that if the structural element is symmetric, and the centre is in the midpoint of the element, then $\hat{\mathbb{S}} = \mathbb{S}$.

Figure B.22 shows a structuring element from a theoretical representation with its centre, and its representation as a binary kernel $\mathbb{S}$. It also shows the two basic operations taken from section A: reflection ($\hat{\mathbb{S}}$) and translation ($\mathbb{S}_{x,y}$) which allow for the kernel to be moved along the binary image. In this work, the structuring elements will be square matrices of odd sizes, $2n + 1$, and the centre will be the element at position $(n + 1, n + 1)$.

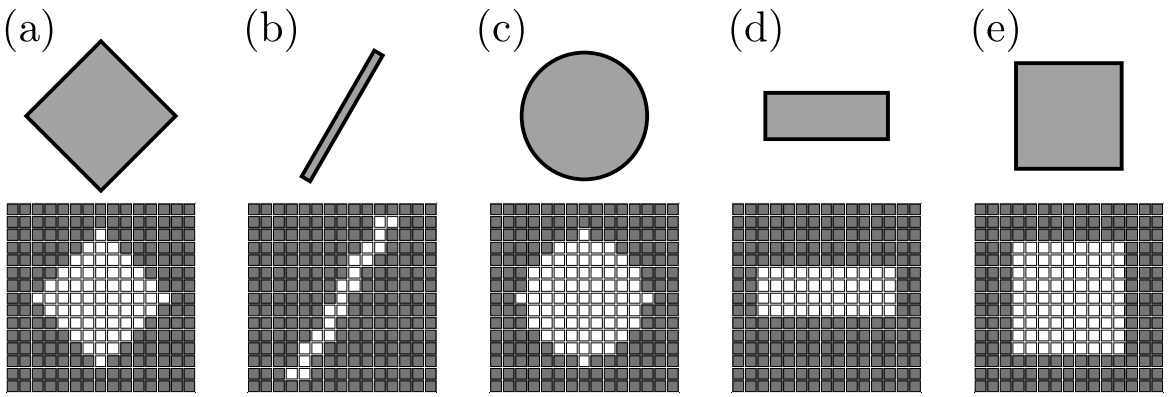


Figure B.23 Examples of structural elements. Representations of structural elements in abstract form (top) and represented as discretisation in matrices (bottom). (a) Diamond. (b) Line. (c) Circle. (d) Rectangle. (e) Square.

Morphological operations

In most cases, morphological operations are performed similarly to convolution, as seen in figure B.6, as a structuring element will be translated along a binary image, performing set operations as it moves. In this section, the following operations will be defined: (i) erosion, (ii) dilation, (iii) opening and closing; as well as some algorithms like (iv) boundary extraction and (v) hole filling. For the coming definitions, let $\mathbb{I}$ be an image and $\mathbb{S}$ a structuring element of size $2n+1$, with a centre at position $(n+1, n+1)$. For simplicity, assume that $\mathbb{I}$ contains only one set —or detected object— in it. The operations will result in new sets, described by the elements in it. In terms of images, the positions that belong to a particular set will have a value of 1 in the resulting image.

Erosion. The erosion of an image can be interpreted as an operation in which the chosen structural element will reduce in area the regions present in the image. The operation is useful in some cases, when disjoint regions corresponding to the same object to be joined together. The erosion of $\mathbb{I}$ by $\mathbb{S}$ will be represented by the symbol $\mathbb{I} \ominus \mathbb{S}$ and defined as

$$\mathbb{I} \ominus \mathbb{S} = \{(x, y) | \mathbb{S}_{x,y} \subseteq \mathbb{I}\}\,. \tag{B.16}$$

Equation (B.16) can be read as *the set which contains all points* (x, y) *in which the translated kernel* $\mathbb{S}$ *is fully contained in a set within* $\mathbb{I}$. A diagram to illustrate this procedure can be observed in figure B.24, which shows a diagram of the erosion process.

Dilation. The operation of dilation can be interpreted as *expanding* the area of a region. In some cases, it allows for disjoint areas to be joined to others. Represented by the symbol $\mathbb{I} \oplus \mathbb{S}$, the dilation the image $\mathbb{I}$ by the element $\mathbb{S}$ is given by

$$\mathbb{I} \oplus \mathbb{S} = \left\{(x, y) | \hat{\mathbb{S}}_{x,y} \cap \mathbb{I} \neq \emptyset\right\}, \tag{B.17}$$

the set can be described as the positions (x, y) in which the translated and reflected element $\hat{\mathbb{S}}_{x,y}$ has a non-empty intersection with the set in the image $\mathbb{I}$. Figure B.24 displays the way erosion and dilation from a structural element $\mathbb{S}$ into a set with disjoint regions $\mathbb{A}$. The structural element present in the picture will be translated throughout the plane containing the set $\mathbb{A}$, and the output produced will follow equations (B.16) and (B.17). Then, figure B.25 show the output of performing the corresponding operations by using different structural elements. The image where the operations are

performed correspond to the binarisation by hysteresis thresholding of a section in a macrophages frame.

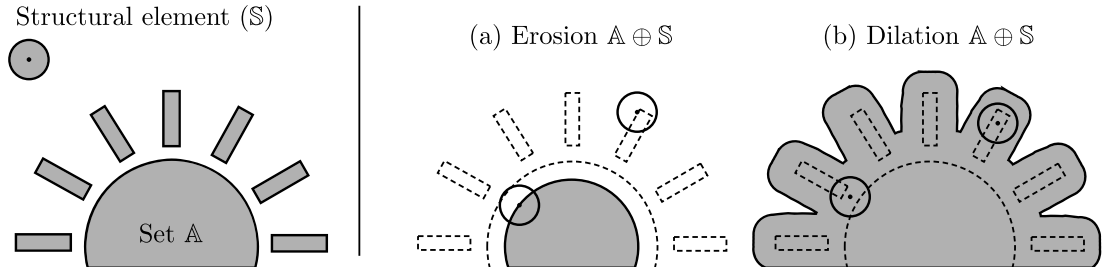


Figure B.24 Diagram of erosion and dilation operations. The set $\mathbb{A}$, is represented by different disjoint regions, while the structural element, $\mathbb{S}$, is represented as a circle, with its centre in its midpoint. The outputs of erosion and dilation are shown in (a) and (b). In erosion (a), only positions where the contention $\mathbb{S}_{(x,y)} \subset \mathbb{A}$ is proper. In dilation (b), as long as the intersection is non-empty $(\mathbb{S})_{(x,y)} \cap \mathbb{A} \neq \varnothing$.

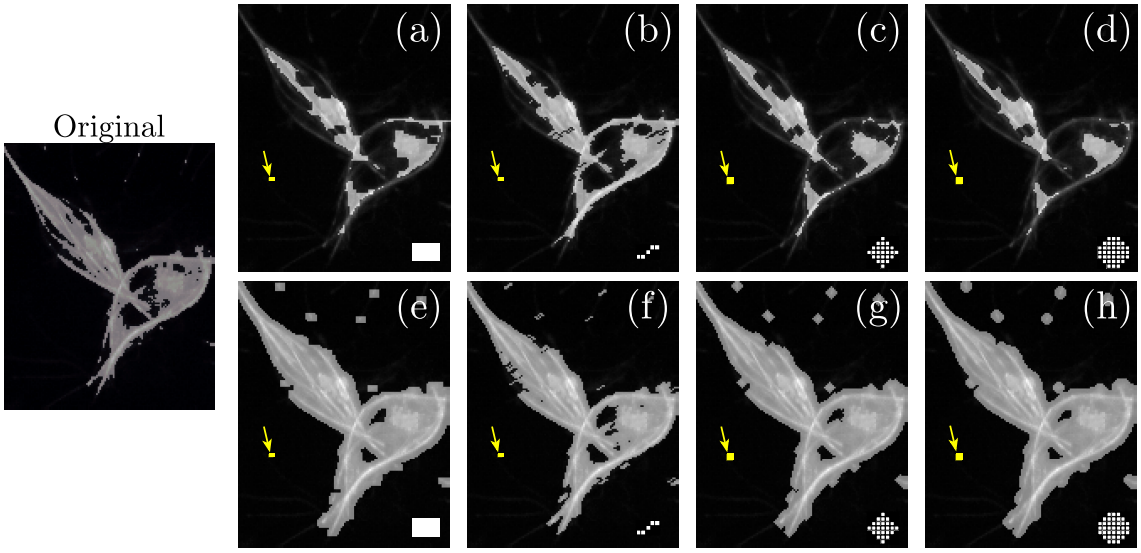


Figure B.25 Example of erosion and dilation outputs based on the different structural elements shown in B.23. All images present a binary image overlapped on a section of a frame in the macrophages dataset. The actual size of each structural element is highlighted in yellow, and marked with arrows, and the shape of the structural element kernel $\mathbb{S}$, is shown amplified at the lower right corner of each image. The output of a binarisation of a macrophages dataset is shown at the far left (Original). **Erosion** is shown in (a-d), and **dilation** is shown in (e-h). The different structural elements used are: a rectangle (a,e); a line (b,f); a diamond (c,g) and a circle (d,h).

Complex operations can be implemented from the previously described ones, as seen in figure B.25, one operation on its own might not provide optimal results, however,

combinations of the techniques could be useful. For example, dilation is useful to join two separated parts of a same object while erosion is useful when eliminating granular noise. Such complex operations are **opening** and **closing**.

Opening and closing. Both operations combine erosion and dilation, in different order, and are used to produce a smoothening effect in the set in the image. The difference lies in the order the operations are performed. Opening is represented by the symbol $\mathbb{I} \bigcirc \mathbb{S}$ and involves an erosion of $\mathbb{I}$ by $\mathbb{S}$, followed by the dilation of the result by $\mathbb{S}$,

$$\mathbb{I} \bigcirc \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}. \tag{B.18}$$

Closing is the analogous operation to opening, in which a dilation is followed by an erosion. The operation is represented by the symbol $\mathbb{I} \odot \mathbb{S}$,

$$\mathbb{I} \odot \mathbb{S} = (\mathbb{I} \ominus \mathbb{S}) \oplus \mathbb{S}. \tag{B.19}$$

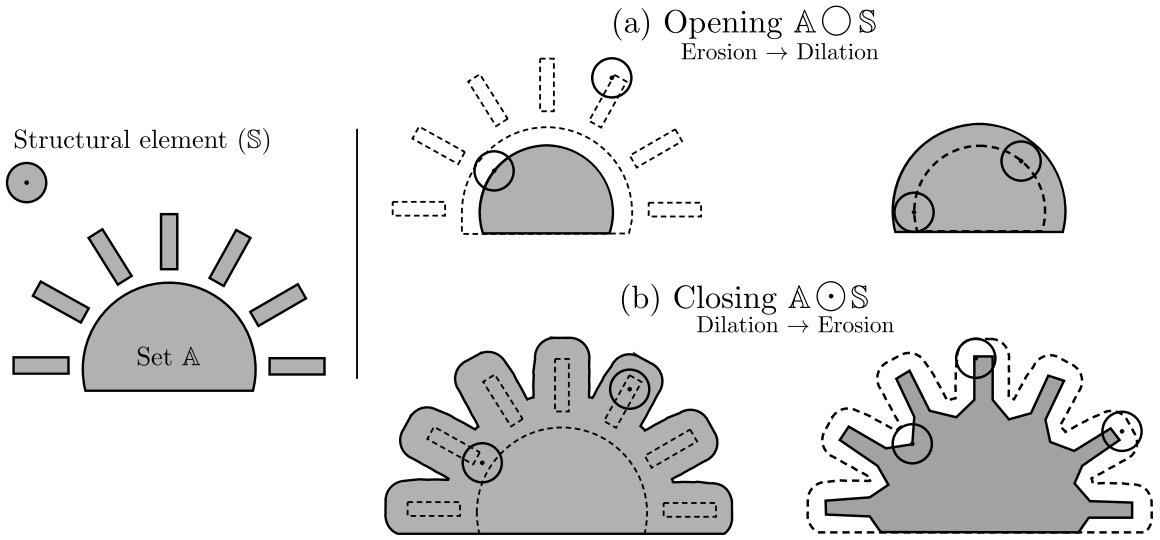


Figure B.26 Diagram of morphological opening and closing operations. (a) shows the process of **opening**, where an erosion is performed to eliminate the sections with a small area and then a dilation to increase the size of the remaining set. (b) shows the process of **closing**, where a dilation allows sections to be joined and then an erosion reduces the overall size of the shape.

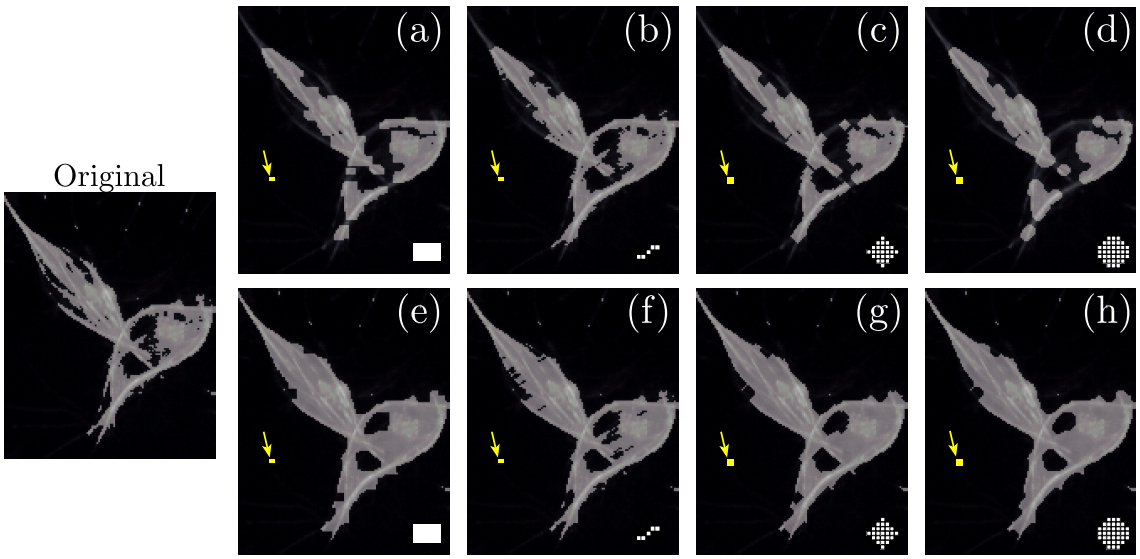


Figure B.27 Example of opening and closing performed using different structural elements on B.23.

Feature extraction

Another useful way to analyse images, involves extraction of certain characteristics in the image called features. Feature extraction is common in problems of texture analysis, where detection of objects is based on characterisation of regions in an image by the variations of intensity values of an image, sometimes modelled through its statistical properties. Texture analysis provides a quantitative description of qualities —such as *smooth* or *rough*— in image regions.

Features are extracted by filtering an image with a number of specific kernels to produce different related images, $\{\mathbb{I}_q\}_q$, all images contain information extracted locally by the different kernels applied. Thus, per pixel, a vector of length q holds local information extracted from the local characteristics of the image. Each pixel could then be grouped with similar pixels through the similarity of the feature vector extracted. As a continuation of the data exploration, two techniques are considered to the analysis, Gabor filters [86] and SIFT/SURF techniques, [87, 88].

Gabor filters

Gabor filters [86] refer to kernels which —as seen in the filtering section of this work— assign to each pixel local information about it. The kernels have a specific orientation and frequency, or wavelength. A common application for texture classification [86]

involves using a bank of filters with varying frequencies and orientations to extract a large vector which contains different information on each pixel, and then using a statistical technique such as k-means [85] to cluster them.

Implementation. Figure B.28 show an example of a standard Gabor filter bank, in which the variation of wavelength and orientations can be observed. The orientations shown are increments of 30 degrees from 0 to 150. The wavelengths were chosen in increasing powers of two, starting at $\lambda_0 = 4/\sqrt{2}$ and up to the hypotenuse length of the input image, in this case $\lambda_{\max} = \sqrt{N_r^2 + N_c^2}$, as stated in the work by Jain and Farrokhnia [89]. This would produce six wavelengths and six orientations, for a total of $q = 36$ kernels. In figure B.28, only a fraction of such kernels are shown for visualisation purposes with different sizes. Note that the sizes displayed in the figure were chosen arbitrarily, and are not to scale.

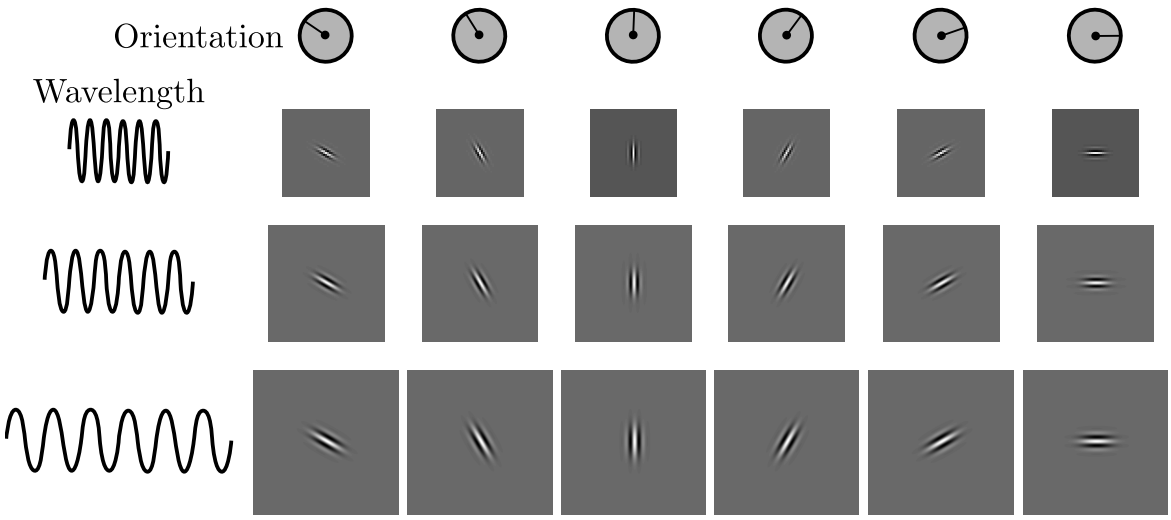


Figure B.28 Example of a Gabor filter bank for feature extraction. Notice the change in orientation of the kernels and the change in size and resolution as the wavelengths increase.

After obtaining the filter bank, each filter is convolved with the frame to produce $q = 36$ images, $\{\mathbb{I}_q\}_q$. Figure B.29 shows the different instances of $\mathbb{I}_q$, after convolving with the Gabor filters in the bank. Notice how each kernel, due to the wavelength and orientation highlights different areas of the original frame.

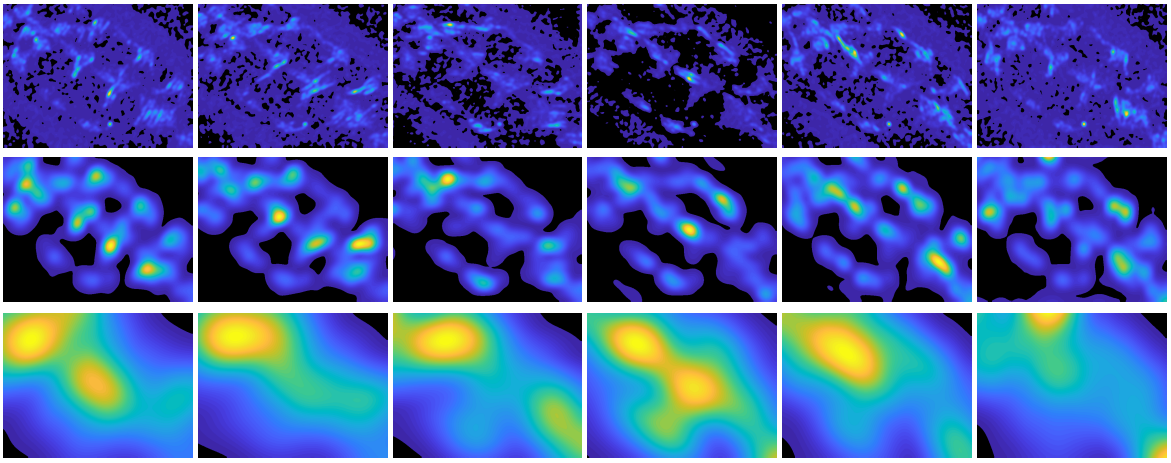


Figure B.29 Convolution output of the different Gabor kernels on the green channel. Each Gabor filter kernel shown in figure B.28 was convolved with the green channel of the test frame. Notice that as the size of the filter increases, due to the wavelength increase, the output of the convolution highlights different areas of the figure of different scales.

Feature vectors are then formed by aligning pixels at same positions in each of the generated images, $\mathbb{I}_q$. Therefore, each pixel in the original image has a vector in $\mathbb{R}^q$ associated to it and the local information from it. K-means [85] is then used to group the different vectors into one of two groups, foreground and background. Figure B.30 represents the output of the experiment.

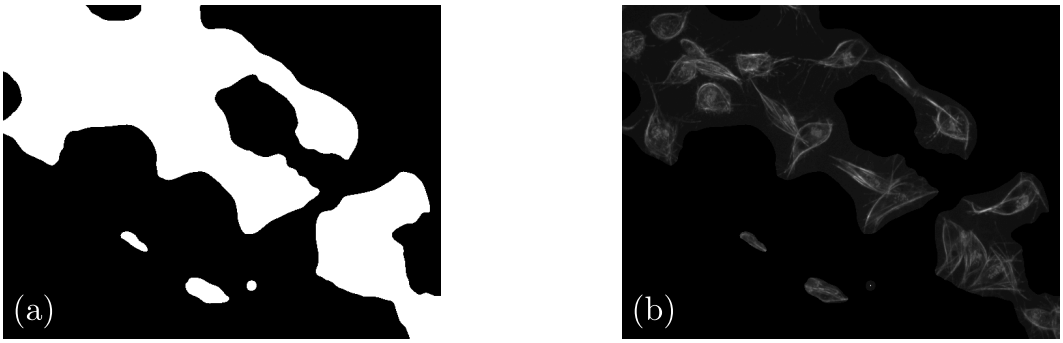


Figure B.30 Foreground and background detected using Gabor features. (a) Represents the binary image of foreground (white) and background (black). (b) Shows the original frame, with the background suppressed for visualisation purposes. Notice that the technique was able to identify areas where cells exist, but the segmentation is not done, as not even single cells were detected.

Scale-Invariant Feature Transform and Speeded-Up Robust Features

In an image, a local feature refers to a pattern of specific structure like a point, edge or a small image patch. The **Scale-Invariant Feature Transform** was introduced

by Lowe [87] as a novel class of local feature detection, which were impervious to transformations such as translation, rotation or even illumination changes. On the other hand, the **Speeded-Up Robust Features** (SURF) [88, 90] were proposed by Bay et al. [90] as a method to obtain a scale and rotation-invariant feature detector that could outperform SIFT.

In this section, both algorithms are described and an illustration is provided to highlight the algorithms' advantages. Finally, an experimental study on the algorithms' ability to follow single cells in consecutive frames is presented.

The SIFT algorithm is centred in finding keypoints in the context of their orientation. The keypoints can be understood as a point on an image chosen by some criterion, in the case of SIFT, local maxima and minima on specific functions of the input image. For it to be robust, it is necessary to guarantee that the image is considered at distinct scales, which is achieved by convolving the input image with Gaussian kernels of varying standard deviation σ. An overview of the algorithm is presented in 4, each step will be explained throughout this section.

Algorithm 4: SIFT FEATURES Produces the SIFT features on image $\mathbb{I}$.

Input: Image: $\mathbb{I}$

1 Construct a scale-space $\mathcal{S}$
2 Laplacian of Gaussians Approximation through Gaussian differences
3 **Finding Keypoints** $\mathcal{KP}$
4 Assigning Orientation to keypoints
5 Generating a **feature**

Let $\mathbb{I}(x, y)$ an image, and let the scale space, $\mathcal{S}(\mathbb{I})$, of the corresponding image be defined by the set of images $\mathbb{L}(x, y, \sigma_r, j)$ where σ_r represents the ith scale and j represents the jth octave. The scale space can be defined with equation (B.21):

$$\mathcal{S} = \{\mathbb{L}(x, y, \sigma_r, j) | \sigma_r = k^r \sigma_0, j = 1, \cdots, N_o\} \tag{B.20}$$

$$\mathbb{L}(x, y, \sigma_r, j) = \text{imresize}(\mathbb{I}, 2^{-j}) \circledast G(x, y, \sigma_r), \tag{B.21}$$

where $G(x, y, \sigma_r)$ is a Gaussian kernel with zero mean and standard deviation $\sigma_r = \sigma_r = k^r \sigma_0$, the function imresize$(\mathbb{I}, 2^{-j})$ produces what Lowe [87] refers to as octaves, which are reductions of the image $\mathbb{I}$ by a factor of $1/2^j, j = 0, 1, \cdots, N_o$. Generating the scale space will produce a set number of scales, N_s and a set number of octaves N_o. Normal parameters used for this include starting with $\sigma_0 = \sqrt{2}/2$ and $k = \sqrt{2}$. The

values for the scale space can be seen in Table B.3, it is common to have $N_s = 5$ scales and $N_o = 4$ octaves.

Table B.3 Choice of parameters for the generation of a scale space.

	Scale →				
Octave	$\sqrt{2}/2$	1	$\sqrt{2}$	2	$2\sqrt{2}$
↓	$\sqrt{2}$	2	$2\sqrt{2}$	4	$4\sqrt{2}$
	$2\sqrt{2}$	4	$4\sqrt{2}$	8	$8\sqrt{2}$
	$4\sqrt{2}$	8	$8\sqrt{2}$	16	$16\sqrt{2}$

The Laplacian or second order derivatives of the images in the scale space is a technique to highlight edges and corners on the image. However, this operation is computationally intensive. The second step in Algorithm 4 involves approximating the Laplacian of Gaussians by calculating differences in consecutive scales within the same octave. Given the $N_s = 5$ scales and $N_o = 4$ octaves, the differences of Gaussians will produce four differences per octave. The process is known as finding the differences of Gaussians (DoG).

Finding the Keypoints In the image, detecting the keypoints involves finding the local maxima and minima in the DoG. A local maxima of a function $f(x, y)$ is a point $(x^\star, y^\star)$ in which $f(x^\star, y^\star) \geq f(x, y)$ for all points in a ball or radius M to the point $(x^\star, y^\star)$, that is, $B_M(x^\star, y^\star) = \left\{(x, y) | (x - x^\star)^2 + (x - x^\star)^2 < M\right\}$. In this, the ball $B_M(x^\star, y^\star)$ represents a neighbourhood. A local maximum is a concept which can be understood intuitively in the context of images, as a point which has a intensity value than some of its neighbours. Particular to the SIFT algorithm, the criterion for determining whether a pixel is considered a local maximum involves the two adjacent octaves, in 3×3 windows. Points with low contrast are then removed from the keypoints, which involves a threshold of intensity per keypoint. Finally, the orientations of each keypoints is obtained by finding the most prominent orientations around the keypoint.

Generating features. Each keypoint is assigned a 16×16 window, subdivided into sixteen 4×4 subwindows. Each of which contains 16 pixels whose orientations are allocated into eight bins in a histogram. This produces a vector in $\mathbb{R}^{128}$ which characterises each keypoint uniquely.

The SURF algorithm uses the same idea as the SIFT algorithm, however it applies key changes to the implementation of its parts in order to improve performance and running time. While SIFT calculates image pyramids and then computes the DoG, SURF creates a stack of images without downsampling which combined with the use of integral images, filters the stack using a box filter approximation of the second order derivatives of Gaussians, which allows for computation of rectangular filters in near constant time [91].

Implementation. A normal implementation which exploits the features created by SURF, involves two images, a reference one which contains the object of interest and a secondary image which contains the object in a particular *scene*. The SURF algorithm would be run on both images and two sets of features would be found. Since SURF detects features regardless of transformations, the same features would be found in both images, and the feature vectors could be compared between them, matching points in the original image and the *scene*. Two tests implementing the SURF features were performed, first on test images of books and then on the images of cells in the `MACROS1` dataset.

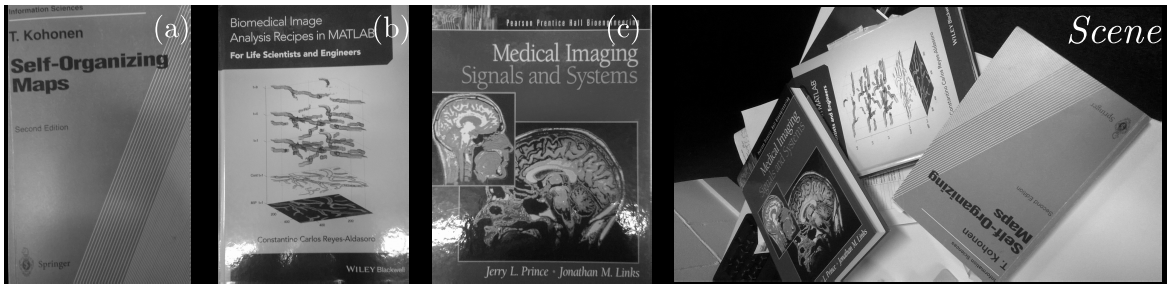


Figure B.31 Setup of generic SURF implementation. (a-c) Show the reference images of the book covers. On the right, the scene with the books. Each book has a collection of SURF features, and the features from each image can be compared to the features detected on the scene.

In the figure B.31, the three books in the experiment are shown photographed showing the full front cover, and a separate scene in which all books are present at different angles, scales and shears. Some parts of the books are overlapped in the scene. The SURF algorithm is run on each of the images, producing separate collections of SURF keypoints. The same algorithm produces vectors associated to each point in each image called features. Each feature corresponds to a vector, which provides a signature of the point in the image. Thus, the features from a reference image can be matched

to the features in the scene; and based on the similarity of them, the points can be classified as strong, when the similarity is high or low when they are different. Finally, A certain number of the strongest features is selected. The output of the experiment is shown in figure B.32. Notice how some of the points in the reference images are matched to points which do not correspond in the scene. This simple analysis, even before analysing images of cells showcases a limitation.

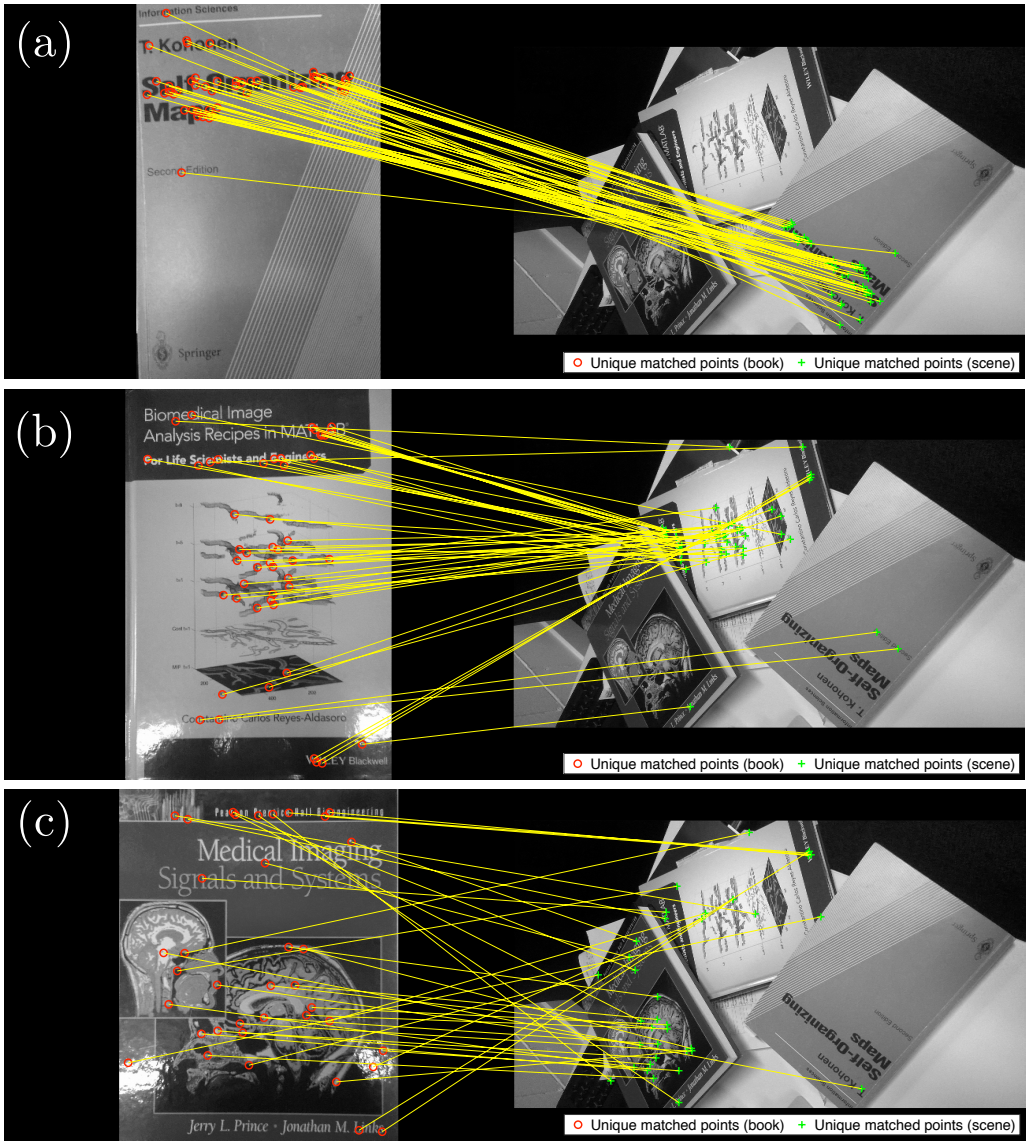


Figure B.32 Illustration of SURF features detection and matching. On the left of each image, three books were photographed. The SURF features were detected on each and then matched to the SURF features in a scene which includes the books. In each case, the 50 strongest matching points were plotted in the single book and matched with their corresponding point in within the scene.

B.2 Machine Learning Techniques

Machine learning techniques are a set of methods for the automated analysis of structure in data[117]. For this work, an example of supervised learning and one of unsupervised were studied. In particular, the Self Organising Maps (SOM) by Kohonen [105] is presented as an example of an unsupervised approach to machine learning. On the other hand, the supervised learning technique reviewed in this section is the Convolutional Neural Network, with the classification of each of the pixels as expressed by Su et al. [118] and Giusti et al. [119] fast-scanning neural networks was of particular interest to this work.

Self Organising Maps A self-organising map (SOM), proposed by Kohonen [105], is a type of unsupervised neural network in which features of an image, such as the intensities, are related to their spatial context. In image analysis, self-organising maps have been successfully used for image segmentation, like in the work by Reyes-Aldasoro and Aldeco [120]. The methodology is simple, as it takes a network —or graph— in which the nodes have some feature as well as spatial coordinates. The network is defined in various topologies depending on the edges connecting the nodes. The most common topologies include grids, rings or lines (figure B.33). For image segmentation, the network will be initialised somewhere in the image, and the network's organise itself to match the input.

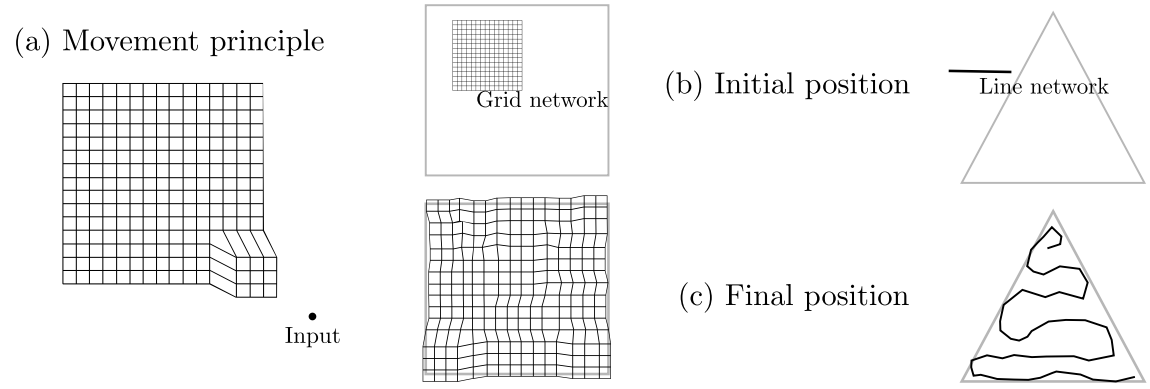


Figure B.33 Common network topologies used for SOM.

The algorithm is run sequentially, taking different pixels and their positions in the input image, $(x, y, \mathbb{I}(x, y))$. The basic principle takes the node closest to the input and moves in response. The movement is normally attraction, however it can involve repulsion of the node with the input particles. As the node moves, a subset of nodes

connected to it, moves as well, the movement can follow certain rules, such as maximum step size. After enough input points have been presented to the network, it will take the desired form. Finally, since the distance function between the network and the input can be determined by the user, customised functions can be implemented to include the information of the node apart from the spatial context, such as the image intensity.

Formulation. In broad terms, let the network $\mathcal{R} = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ are nodes assigned to positions in the plane and $\mathcal{L}$ are some edges linking the some of the nodes in $\mathcal{V}$. The nodes are specified in equation (B.22),

$$\mathcal{V} = \left\{ \mathbf{m}_i = (x_i, y_i) \in \mathbb{R}^2 : i = 1, \cdots, n_v \right\}. \tag{B.22}$$

Let the tuple $(\mathbf{x}_t, \mathbb{I}(\mathbf{x}_t))$ be an input to the SOM at iteration t. Then, the SOM algorithm proposed by Kohonen [105] follows two basic steps: (i) identifying the closest node in the network to the input, shown in Equation (B.23), and (ii) update the positions of the nodes inside a neighbourhood, determined by a distance n_e to the winner node $\mathbf{m}_c$, (B.24),

$$\mathbf{m}_c(t) = \underset{c \in \{1, \cdots, n_v\} S}{\arg\min} \|\mathbf{x_t} - \mathbf{m}_i(t)\|_2^2 \tag{B.23}$$

$$\mathbf{m}_i(t+1) = \begin{cases} \mathbf{m}_i(t) + \alpha_t \left(\mathbf{x}_t - \mathbf{m}_c(t) \right) & , (i,c) \in \mathcal{L} \text{ and } \mathrm{Dist}(\mathbf{m}_i, \mathbf{m}_c) \leq n_e \\ \mathbf{m}_i(t) & , \text{otherwise} \end{cases}, \tag{B.24}$$

where $\mathrm{Dist}(\mathbf{m}_i, \mathbf{m}_j)$ refers to the distance from node i to node j in the shortest path determined by the edges $\mathcal{L}$. The shortest path in the network is determined by the number of jumps from the winner node to its neighbours.

Convolutional Neural Networks Convolutional Neural Networks (CNNs), as a part of the deep learning techniques has become really popular amongst researchers because of the ease of use and relatively easy implementation, like on the work by Jia et al. [121], where a framework for developing production-grade CNNs is explained. As mentioned before, the work performed involved the work by Su et al. [118], that involved the network shown in figure B.34. The input of the network were 28×28 patches, with the pixel of interest being the middle one, which on this work was interpreted as the pixel at coordinates (14,14). As seen on figure B.34, the first convolutional

stage has 12 kernels of size 5×5, followed by a max pooling stage. These steps occur twice, and end in a two hidden layers, fully connected stages that classify the pixel in question. In general, CNNs require a large amount of data, and because of the numbers of parameters in all the layers, the processing of each one has increased significantly.

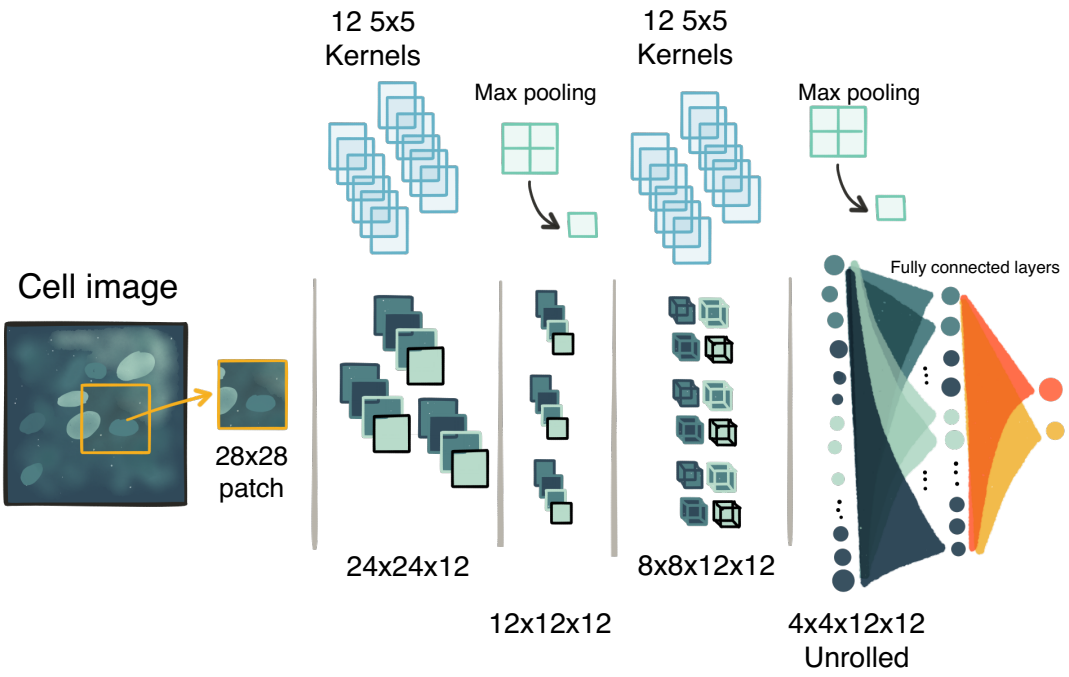


Figure B.34 Layout of the CNN architecture. It has two convolutional stages, each with twelve 5×5 kernels and two max pooling stages. After the second max pooling completed, there are two fully connected neural network layers.

B.3 Segmentation performance metrics

Figure B.35 shows the possible characterisation of the output of a segmentation algorithm, given its output and the value of the same pixel in the ground truth. Four possibilities, depending on the comparisons: (i) **True positive** (TP) as the pixels where both the output and ground truth show a positive or 1; (ii) **True negative**, (TN) in pixels where both images show a negative or 0; (iii) **False positive**, (FP)

where the segmentation shows a 1 but the ground truth shows a 0, and (iv) **False negative** (FN) where the segmentation shows a 0, but the ground truth shows a 1.

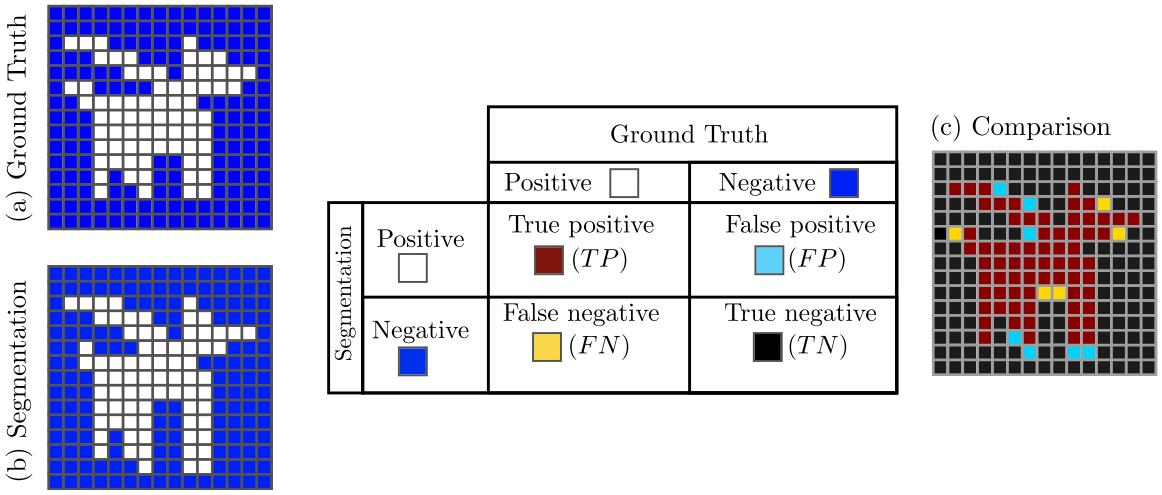


Figure B.35 Diagram of comparison of ground truth (a) against segmentation output (b). (c) Shows in dark reed the pixels which correspond to True Positives or *hits*, in cyan the false positives, in yellow the false negatives and in black the true negatives.

Counting the number of pixels that correspond to each category can produce metrics for the performance of a segmentation, or how close the automatic output is from a given manual annotation. In the works by Jaccard [73], Fawcett [74] and Randen and Husoy [75], such measures are provided. First, the Jaccard similarity index [73], or simply Jaccard index, is defined in terms of sets and their sizes, as the ratio of the intersection by the size of the union. In terms of the statistics described before and in figure B.35, it can be computed by the number of true positives divided by the sum of true positives, false positives and false negatives.

On the other hand, in the work by Fawcett [74], two measurements are presented. First of all, **precision**, which measures how many detected pixels are relevant, by computing the ratio of true positives and the overall number of detected pixels, $TP+FP$. Second of all, the work presents the **recall** measurement, which computes the ratio of true positives with the sum of the true positives and the false negatives —or points that should have been detected, but were missed—, this measurement can be seen as a ratio of the relevant pixels that were detected.

Finally, Randen and Husoy [75] present **accuracy** or **misclassification** which compares the number of correct detections, whether positives or negatives against the overall number of pixels. Equation (B.26) shows all measurements previously described,

and can be observed in figure 2.26,

$$\text{jaccard} = \frac{TP}{TP + FN + FP}, \qquad \text{precision} = \frac{TP}{TP + FP}, \tag{B.25}$$

$$\text{recall} = \frac{TP}{TP + FN}, \qquad \text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{B.26}$$

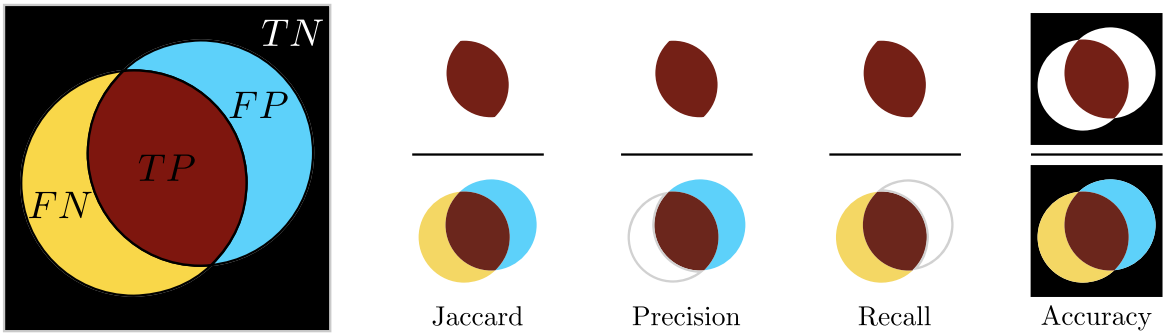


Figure B.36 Graphical representation of the Jaccard index, Precision and Recall measurements. The diagram shows an abstract representation of the sets of pixels in an image, and how they can be classified. The ratios described in equation (B.26) is shown, based on this graphical representation.

Appendix C

PhagoSight and the Keyhole tracking algorithm

The PhagoSight software [17, 59, 102] is a semi-automatic tool originally developed for the analysis of fluorescent neutrophil and macrophage migration in a zebrafish model. The tool is released open-source as a collection of MATLAB® m-files that can be called from the command-line or with a Graphical User Interface (GUI). PhagoSight is a framework for cell tracking that uses the Keyhole algorithm and hysteresis thresholding based on Otsu [81] for the segmentation.

C.1 Segmentation.

The segmentation stage occurs by taking a modified version of the Otsu [81] algorithm that consists of finding a single threshold $k^\star$, and defining the two levels (k_0, k_1) shifting in both directions, where $k_0 = 0.95k^\star$ and $k_1 = 1.05k^\star$. Then a hysteresis process is done. The software allows the user to perform a manual modification of the thresholds through a graphical user interface GUI, figure C.1. As post-processing, a parameter is set to discard any regions below a given area. The parameter can either be set by the user or taken as one quarter of the distance between the mean area and three standard deviations. That is, if the mean of the areas detected by the segmentation is μ_B, and the variance σ_B^2, then the objects that will be discarded will be those that are less than $(\mu_B - 3\sigma_B)/4$. All the objects detected in the binary images within a dataset are labelled and saved to disk.

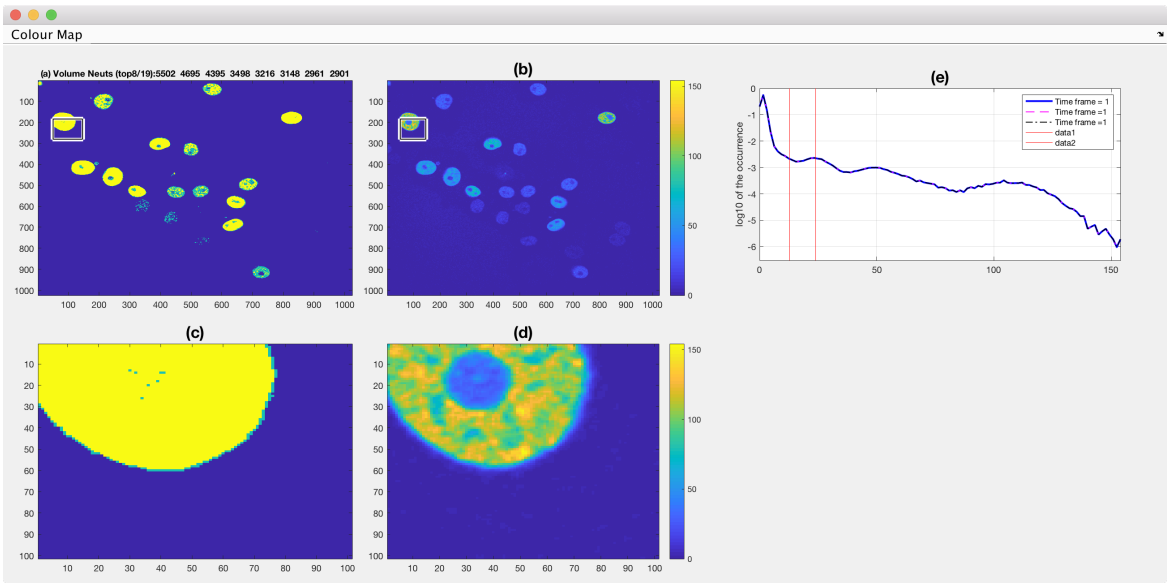


Figure C.1 PhagoSight user interface for selecting manual thresholds. This interface allows the user to verify on the thresholds that were selected automatically and optionally change them. A view of the classes defined by the thresholds selected is shown in the first image of the dataset (top left), the original image of the dataset is shown (top middle) and a view of the histograms corresponding to the first, middle and last frames is shown with the positions of the thresholds (top right). On the bottom row, a zoom into the image is shown, for the user to have a more detailed view of a given area of interest. The example shows the ISBI N2DH-GOWT1 dataset.

C.2 Implementation of automatic thresholds for segmentation module

PhagoSight allows for user modification of the thresholds used for the hysteresis. In the context of the Cell Tracking Challenge (CT-Challenge), a fully automated software was requred, which

Let (lt_{otsu}, ht_{otsu}) be the original Otsu-based thresholds generated automatically by Phagosight on an image $\mathbb{I}$, and let $M_{\mathbb{I}} = \text{mode}\{\mathbb{I}\}, m_{\mathbb{I}} = \min\{\mathbb{I}\}$. Then, the new automatic thresholds (LT, HT) will be produced by the introduction of parameters (α, β) as seen in equation (C.2).

$$LT = M_{\mathbb{I}} + \alpha\left(lt_{otsu} - m_I\right), \tag{C.1}$$

$$HT = M_{\mathbb{I}} + \beta\left(ht_{otsu} - m_I\right). \tag{C.2}$$

The parameters were optimised for the Challenge implementing the Particle Swarm Optimisation algorithm [112], with the objective function taken as the sum of the evaluation codes for segmentation and tracking provided by the Challenge organisation, that is:

$$\max_{\alpha,\beta} \mathrm{SEG}(\alpha,\beta) + \mathrm{TRA}(\alpha,\beta), \tag{C.3}$$

where SEG and TRA are functions which evaluate the segmentation and tracking results against the ground truth and give a value between 0 and 1. The algorithm was chosen because it does not make any assumptions about the nature of the objective function. The only modification made to the algorithm were made to accommodate for feasibility in the particles generated, for instance, the levels had to be forced into the $0 < LT < HT$ interval. It was achieved by discarding any bad particle and generating a new random one, which did not seem to have a negative effect on the results. The parameter `minblob`, used to remove objects that were too small, was chosen. The value would disregard objects with a small area a as noise. The discriminating size was computed by the formula $\mathrm{minBlob} = \min\{60, (\mathrm{mean}(a) - 3\,\mathrm{std}(a))/4\}$, where minBlob is the discriminating value for the objects and 60 was chosen empirically.

C.3 Keyhole Tracking Algorithm.

The tracking process is done with the segmented images using the keyhole tracking algorithm [101, 59] described in the previous section. As the objects have been detected and labelled, the task of the tracker involves the generation of a correspondence between the objects' labels in the images with the tracks.

The keyhole tracking algorithm is a technique that links the segmented objects in contiguous frames through the analysis of the velocity and direction of the object at frames $t-1$ and t and estimating the position of the object at frame $t+1$. The methodology assumes that the most likely position of the child object, *i.e.* the one at $t+1$, will follow on the same direction and velocity of the object at present frame, t, and parent, object at $t-1$. As changes in the velocity and direction are possible, the algorithm generates two regions of probability set to anticipate for the child object's landing in frame $t+1$.

The name of the algorithm follows the shape of the probability regions defined, which together resemble an old style keyhole. The distance between the parent and present objects, l, is used to define two regions: the *wedge*, which is an arc of length

$\theta_w = \pi/3$ and radius $r_w = 3 \times l$ to accommodate for objects accelerating; and the *circle*, which accounts for the rest of the $\theta_c = 5\pi/3$ arc not covered by the wedge, and that has a radius of $r_c = l/2$. Figure C.2 shows the different regions defined previously. The algorithm starts by defining just a circular region $\theta_w = 2\pi$ and no keyhole since no information on movement is available at the moment. The radius of the circle r_c is defined by half of the minimum distance between objects.

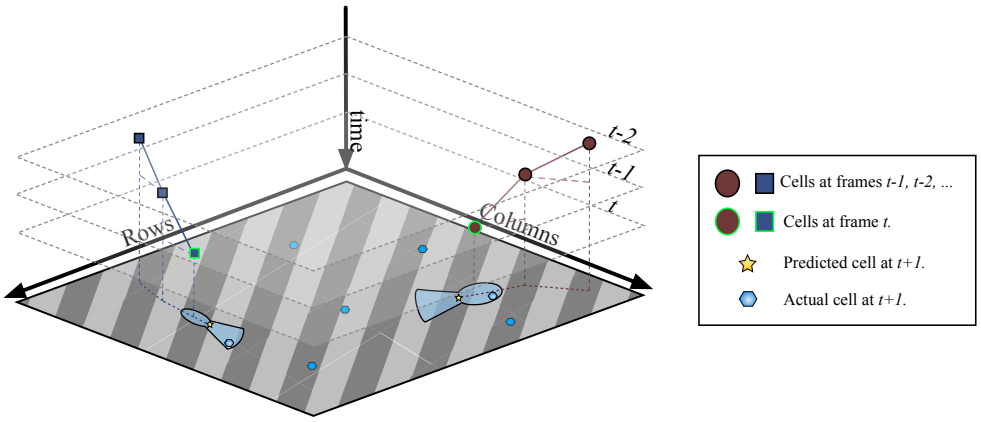


Figure C.2 Keyhole algorithm explanation. Four different time frames are represented vertically, from top to bottom: starting at $t-2$, then the previous time frame $t-1$, the present frame t and the upcoming frame $t+1$ at the bottom. Two examples of different velocities are shown, cells represented with a blue square $\square$ and a maroon circle $\circ$. A landing position is predicted taking into account the velocity of the previous frame; and to account for changes in direction, the keyhole algorithm incorporates the information from the position at $t-1$ and the position at t, to determine the size of the frontal *wedge* and circle.

The output of the PhagoSight framework consists of a large table containing information from each of the tracks detected, given a particular segmentation. Table C.1 shows the output information from the PhagoSight framework. It is important to notice that PhagoSight does not account for overlapping objects or cellular processes such as mitosis, thus the one of the two cells generated would be regarded as a new cell and a new track identifier would be assigned to it. Finally, the PhagoSight framework includes a set of tools to amend and manually edit the tracks, either in code or via a GUI.

Table C.1 Track information retrieved from PhagoSight framework. The information utilised for the analysis is highlighted: position provides the coordinates x_t, y_t, z_t of the cell in the image per frame t, time frame is the temporal position in the dataset, t, label in binary image refers to the label assigned to the cell at segmentation and final track label corresponds to the track identification, $\mathcal{T}$.

Position (x_t, y_t, z_t)	Distance to closest	Time frame
ID	Parent cell	Child cell
Velocity	Volume	Label in binary image
Keyhole	Track ID	Final track label
Bounding box information	Volume to surface	Sphericity
Neighbours at different distance brackets	Distance to disappear	Distance to appear

Appendix D

Introduction to partial differential equations in image analysis

In this appendix, an overview of partial differential equations in image analysis. In chapter 3, active contours level set methodologies are employed to analyse the data. To provide some context, a brief summary of the mathematical theory of such techniques in the context of image analysis is presented. The theory is taken from the classical work by Sapiro [98].

D.1 Partial differential equations in image analysis

Considering images as discretisations of continuous objects or signals, partial differential equations (PDE) in image processing is the result of infinitesimal operations done through numerical analysis. The following convention will be made for the representation of a partial differential equation of a given curve c, any of the symbols in Table D.1 will be used throughout this Section.

Table D.1 Different notations used to represent partial differential equations.

Original curve	Notation 1	Notation 2	Simplified notation
$c(s)$	$\partial c/\partial s$	$\partial_s c$	c_s

Let $c(p) \subset \mathbb{R}^2$ be a planar, parametric, closed curve, *i.e.* $c(p) = \{x(p), y(p)\}$, with parameter $p \in [0, 1]$ and $c(0) = c(1)$, then the tangent $\mathbf{t}$ to c with respect to p, is

represented by equation (D.1),

$$c_p = \frac{\partial c}{\partial p} = \left(\frac{\partial x}{\partial p}, \frac{\partial x}{\partial p}\right)^T = (\partial_p x, \partial_p y)^T, \tag{D.1}$$

the unit vector will be considered, and for notation the tangent changed to $c_s = c_p/|c_p|$, *i.e.*

$$c_s = \frac{(\partial_p x, \partial_p y)^T}{\|(\partial_p x, \partial_p y)^T\|_2} \tag{D.2}$$

Equation (D.2) refers to a unit vector giving a $|c_s| = 1$. Such parameter s which produces a unit sized tangent vector c_s will be called the **arc length**. Being a unit vector, it follows that $|c_s|^2 = \langle c_s, c_s \rangle = 1$. As it is equal to a constant, its derivative would be zero.

$$\begin{aligned} \langle c_s, c_s \rangle &= 1 \\ \frac{\partial \langle c_s, c_s \rangle}{\partial s} &= 0 \\ \langle c_s, \partial_s c_s \rangle + \langle \partial_s c_s, c_s \rangle &= 0 \\ 2\langle c_s, c_{ss} \rangle &= 0 \end{aligned}$$

The tangent is a vector perpendicular to the normal $\mathbf{n}$, furthermore, the curvature is defined by the second derivative of the equation with respect to the curve. Since it has been shown that $\langle c_s, c_{ss} \rangle = 0$, then the curvature will be proportional to the normal at a given point, s, in the curve:

$$c_{ss} = \frac{\partial^2 c}{\partial s^2} = \kappa \mathbf{n},$$

where $\kappa = |\partial_s^2 c|$.

Definition 8 (Tangent and Normal) *Let $c(p)$ a closed curve, then the **tangent** and the **normal** to the curve are defined by*

$$\mathbf{t} = c_s = \frac{c_p}{\|c_p\|} \qquad c_{ss} = \kappa \mathbf{n},$$

where $\kappa = \|c_{ss}\|$ is defined as the curvature.

As mentioned before, curves represent boundaries of objects, in which the arc length, s, and curvature, $\kappa = |C_{ss}|$ have been defined. These two properties of the curve are relevant because they are invariant to linear transformations. Linear transformations can be affine or euclidean, and are described in Section A.2. In this section, a brief

summary of partial differential equations (PDE) in image analysis is outlined as an introduction to curve evolution algorithms such as active contours and level set methods.

D.2 Curve Evolution.

Generally, in curve evolution, all the concepts presented before of a curve and its derivatives are introduced to the notion of **time**. This approach allows to model different behaviours of the curves which cause them to evolve and change their shape.

For every curve c, let $\vec{\mathbf{v}}$ be the vector that describes its velocity through the partial differential equation (D.3)

$$\frac{\partial c(p)}{\partial t} = \vec{\mathbf{v}}(p,t). \tag{D.3}$$

Equation (D.3) show how at every point, and depending on the velocity, the curves might change, and deform in time. As seen in the diagram in figure D.1, at each point the tangent and the normal vectors can be expressed as a coordinate axis of the velocity vector. However an interesting property is the velocity $\vec{\mathbf{v}} = c_t$ is such that the tangential component does not affect the geometry of an evolving curve, Equation (D.4).

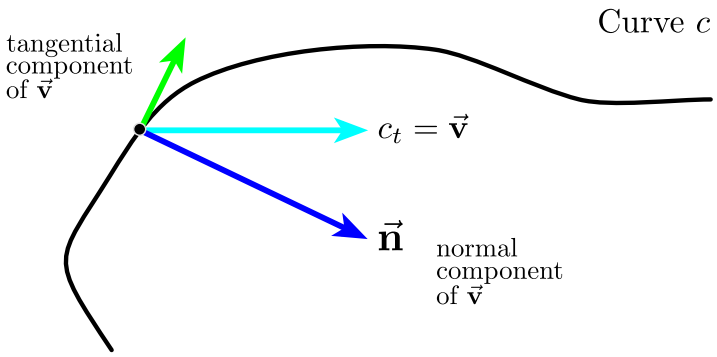


Figure D.1 Tangential and normal components to the velocity of a curve. The tangential component to the velocity vector will only influence the speed at which a particle travels throughout the curve. The normal component to the velocity $\mathbf{n}$, will deform the curve, property of interest in curve evolution algorithms.

$$c_t = \vec{\mathbf{v}} \quad \Rightarrow c_t = \langle \vec{\mathbf{v}}, \mathbf{n} \rangle \, \mathbf{n} \tag{D.4}$$

As the tangent does not affect the shape of the curve, the velocity vector will be a projection along the normal. Thus, in order to model the motion of the curve in time, the value of such projection $V = \langle \vec{\mathbf{v}}, \mathbf{n} \rangle$ will be the starting point to a model. Figure D.2 shows a diagram of examples of types of motion modelled through equation (D.4).

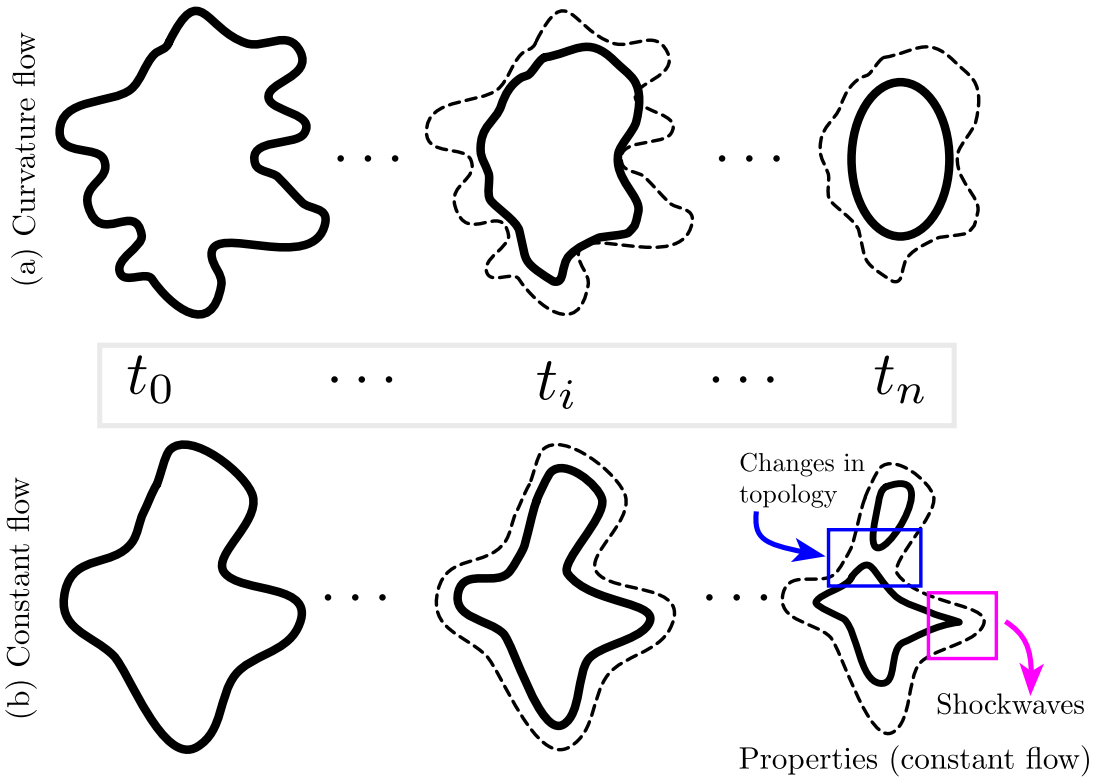


Figure D.2 Representation of two iconic curve evolution models: (a) Curvature flow and (b) Constant flow. In both curves, three stages of the development of the curve can be observed $t_0 < t_i < t_n$. The solid lines represent the curve at each time stage, whereas the dotted line (- -) represent the previous state of the curve, for visualisation purposes.

Implicit functions

In the context of Level Set Methods, curves are modelled in terms of implitic functions.

Definition 9 (Tangent and Normal with implicit function) *Let ϕ be an implicit function to a curve, then the normal $\vec{N}$ and the tangent $\vec{T}$ of the curve are defined by:*

$$\vec{N} = \frac{-\nabla\phi}{|\nabla\phi|}, \qquad \vec{T} = \frac{-\overline{\nabla\phi}}{|\nabla\phi|}, \tag{D.5}$$

where $\overline{\nabla\phi}$ is the conjugate to $\nabla\phi$.

Furthermore, the curvature κ of the curve is defined as

$$\kappa = \text{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right), \tag{D.6}$$

where $\text{div}(\cdot)$ is the divergence. Including the definition in equation (D.4) taking into consideration the implicit function ϕ. In order to move the curve, the level set $\phi(s) = 0$ must be moved based on a velocity V based on equation (D.11):

$$\phi_t = V|\nabla\phi|. \tag{D.7}$$

Curve Evolution Algorithms

Curve evolution algorithms are techniques used for segmentation in which the outline of a detected object is considered as the contour at the level of a 3D function. The two techniques overviewed in this thesis are Active Contours [122, 110], as seen in chapter 5 and Level Set Methods, in the implementation proposed by Lu et al. [99] from chapter 3. As stated in figure D.2, the methods will involve the general equation D.3,

$$c_t = V\mathbf{n}. \tag{D.8}$$

Active Contours. Let $c_t = V\mathbf{n}$ be specified by equation (D.9), where V is going to deform the curve with a certain velocity given by,

$$c_t = V\mathbf{n} = \left[g(x,y)\kappa - \langle\nabla g, \mathbf{n}\rangle\right]\mathbf{n}. \tag{D.9}$$

In equation (D.9), $g(x,y)$ is a **function of the image** $\mathbb{I}$, for example the reciprocal of the gradient $g(x,y) = 1/\nabla\mathbb{I}$. The following conclusions can be drawn from the choice of V:

- If $g(x,y) = 1/\nabla\mathbb{I}$, then at moments with well defined boundaries, g will take a small value, which translates to a smaller velocity V and thus less movement from the curve.
- The term ∇g involves both the reciprocal of the gradient of the image as well as the second derivative of the image, which will promote the stopping point of the velocity even more sensitive to edges, even if they are not as strong.

- Taking the projection of the gradient against the normal **n** because of the nature of moving curves.

From this initial approach, the function V presented would only need slight modifications, such as constant motion **n** to make the movement go faster.

Level set method formulation Consider a closed planar curve $c : S \to \mathbb{R}^2$. The curve can be parametrised, $c(p)$, but another way to represent it is by taking all the points in a function that hold a certain equation to zero.

Definition 10 (Level set) *Let $\phi : \mathbb{R}^2 \to \mathbb{R}$ an implicit scalar function with two variables (x, y) such that $z = \phi(x, y)$. The function ϕ will be understood as implicit, because it depends on the curve c. A curve, understood as a shape in a plane is represented by the level set*

$$c = \{(x, y) \,|\phi(x, y) = c_0\}\,, \tag{D.10}$$

where commonly $c_0 = 0$.

In order to implement curve evolution algorithms, tangent and normal vectors, as well as the curvature must be defined taking the implicit function ϕ into consideration.

Including the definition in equation (D.4) taking into consideration the implicit function ϕ. In order to move the curve, the level set $\phi(s) = 0$ must be moved based on a velocity V based on equation (D.11):

$$\phi_t = V|\nabla\phi|, \tag{D.11}$$

where ϕ_t defines the evolution of the curve as time evolves.

Index

anglegram, 104, 107
    definition, 108
    inner angle, 104
    separation, 106

cell migration, 7
    chemotaxis, 9
    electrotaxis, 9
    haptotaxis, 8
    mechanotaxis, 9
cell motility, 7
    polarisation, 7, 8
    protrusion and adhesion, 8
    retraction, 8
    translocation, 8
cell(s), 7
clump, 77
clump(s), 37
clumps, 4
colour space
    definition, 29
    RGB, 30

Filter, 60, 196
Filters
    Box, 61, 197
    Disk, 61, 197
    Gaussian, 61, 198
fluorescence
    definition, 17
    fluorophores, 17

Green Fluorescent Protein, 19

Image gradient, 65, 202
imaging
    definition, 16

junction
    depth, 106
junction(s), 97
    corners, 99
    peak, 99

macrophages, 10
microscopy, 16
    fluorescence, 16, 17
    methods, 16
model organisms
    definition, 19
Morphology, 71, 216

PDE
    arc length, 240
    curvature, 240
PhagoSight, 129
    definition, 233
    Keyhole algorithm, 235
pixel, 29

Postprocessing, 71, 215

Self Organising Map(s), 228
Self Organising Maps, 228
Set theory
    Universe, 188
SIFT
    Difference of Gaussians (DoG), 225
    octave(s), 224
    scale space, 224
Spatial domain, 55, 183