

Shape Models for Image Segmentation in Microscopy

A dissertation submitted to the University of Dublin
for the degree of Doctor of Philosophy

Kangyu Pan

Trinity College Dublin, September 2013

SIGNAL PROCESSING AND MEDIA APPLICATIONS
DEPARTMENT OF ELECTRONIC AND ELECTRICAL ENGINEERING
TRINITY COLLEGE DUBLIN



To my family, friends, and the past.

Abstract

This thesis presents three model driven segmentation algorithms for extracting different types of microscopic objects based on prior shape information of the objects. The development of these algorithms is motivated by the demands of image processing for facilitating the neurological study of memory formation [46] and the investigation of a novel muscle regeneration technique [40]. This thesis has been divided into three parts, and each part proposes a shape model for segmenting the different image objects in different applications. These models are intended to be designed as fully automatic, however in order to achieve better results for the data in different scenarios some parameters in the models require user intervention.

The first part presents a novel Gaussian mixture shape modelling algorithm for protein particle detection and analysis in the images delivered from the biological study of memory formation. The biological study investigates the roles of the different types of proteins in the protein synthesis governing the formation of long-term-memory. The investigation is based on the colocalization analysis of different proteins in the living specimen which are stained with different fluorescent markers and are imaged separately by using a confocal microscope. However, some protein particles are clustered in the observed images. In order to detect the candidate particles in each cluster-spot (cluster of particles) that has a merged appearance in the images, the outlined shape modelling algorithm fits the intensity profile of each spot of the protein particles (some are cluster-spots) with a 2-D Gaussian mixture model (GMM). The new GMM approach not only estimates the number of candidate particles in a cluster spot, but also parametrizes the shape characteristics of the particles for the later co-localization analysis. Since the optimization process for the parameters of the GMMs is intractable, a novel error-based split-and-merge expectation-maximization (eSMEM) algorithm (as part of the shape modelling algorithm) is applied to estimate the parameters of the mixture components of the model as well as the number of components. The shape modelling algorithm is completed by adding a pre-processing stage that overcomes scanning-noise and background luminance problems from the acquired image data. During the pre-process, a contrast threshold is defined by the user in order to have a initial rough segmentation of the separate spots in the image. The separated spots are then parameterized by the GMMs automatically. After shape modelling of the particles, the usefulness of the results to the biological analysis is demonstrated by showing how the shape models are used to perform data analysis and co-localization analysis on the protein particles in the images.

The second part presents a wavelet-based Bayesian segmentation model to reconstruct the shape the synapses (where the protein synthesis takes place) from a stack of image slices recorded with a confocal microscope. The shape information of the synapses is captured by scanning the synaptic object at multiple different ‘depths’ from top to bottom of the living specimen, and is stores as an image stack of the serially scanned images. The challenges in this work are

the irregular luminance of the synapses and the presence of the ‘out-of-focus’ synaptic features. These are caused by the defect of the fluorescent staining process and the physical limit of the confocal microscope. A Bayesian model is outlined for the segmentation by incorporating the ‘sharpness’ information of the objects, the global intensity histogram, and the inter-slice intensity behaviour. The ‘sharpness’ information is assumed to be luminance invariant and is derived by the high frequency content of the image in the wavelet domain. The global intensity histogram is used to separate the potential object and the background noise. The inter-slice intensity information is the intensity profile across the slices of the image stack at the same pixel site of every slice. The Bayesian segmentation is optimized automatically by using the graph-cut method. Since the determination of out-of-focus synaptic features is subjective, the weights of the likelihoods and the prior in the model are user interactive in order to achieve better segmentation of the 3D synaptic shape.

The last part presents a new active contour mode (Cellsake) for segmenting the overlapped cell/fibre objects in the skeletal muscle images. These images are delivered from the research of a novel muscle regeneration technique in skeletal muscle therapy. In order to validate and evaluate the performance of the new regeneration technique, the cell and fibre (formed by multiple cells) objects need to be detected and classified from the image data collected from experimental samples. The challenge of the segmentation is the high variation in the shapes of the fibre objects the random distribution of the nuclei over the fibre cell-body. More importantly, since the images are 2-D projections of the 3-D test samples, the actual cells/fibres distributed at different depth levels in an observed sample may be projected as overlapped objects in the resulting image. These overlapped objects with various shapes can ‘confuse’ the classification of individual fibres, and the shape estimation of the occluded parts of the objects is difficult. In order to distinguish the fibres from overlapped objects and segment the candidate cell/fibre from each overlapped object automatically, the outlined algorithm divides each separated region (region of interest, ROI, may consists of overlapping objects) in the cell-body image into small ‘cell candidates’ indicated by the separated nuclear regions (clusters of nuclei) in the ROI. Each ‘cell candidate’ is associated with an active contour (AC) model, and the deformation is constrained by the energy terms derived from the shapes of the cells (cellular energy). After the deformation, the ACs are merged when the corresponding ‘cell candidates’ belong to the same fibre. Consequently, the merged shapes become the segmentation results of the cell/fibre objects in the image. In order to achieve the optimal segmentation result for the objects in different scenarios, the weights between the different energy terms in the model are adjustable by the users.

Declaration

I hereby declare that this thesis has not been submitted as an exercise for a degree at this or any other University and that it is entirely my own work.

I agree that the Library may lend or copy this thesis upon request.

Signed,

Kangyu Pan

September 27, 2013.

Acknowledgments

I would like to thank my supervisors, Prof. Anil Kokaram and Prof. Frank Boland for their time, advice and patience. Gratitude must also go to my co-supervisor, Dr. David Corrigan, for all the help and advice he has given, in particular when reviewing my publications and thesis. Thanks must also go to all past and present members of the SIGMEDIA group and the staff in Electronic and Electrical Engineering in Trinity College who have assisted me in my work. I would also like to thank Jens Hillebrand, Prof. Mani Ramaswami and Michael J. Higgins for providing the image data and biological information in this work. This work was funded in part by the Science Foundation Ireland (08/IN.1/12112). Thanks to everyone involved for their foresight. Finally, I would like to thank my family and friends. In particular to my parents, for all their encouragement, understanding and support they given me.

Thank you all very much.

Contents

Contents	v
List of Acronyms	viii
1 Introduction	1
1.1 Spot Analysis	2
1.2 3-D Shape Reconstruction	2
1.3 Segmentation of Overlapped Objects	3
1.4 Thesis Outline	4
1.5 Publications	5
2 Image Segmentation in Microscopy: A Review	7
2.1 A Review of Segmentation Techniques in Microscopy	7
2.1.1 Thresholding	8
2.1.2 Watershed Segmentation	9
2.1.3 Data Clustering	11
2.1.4 Machine Learning Classifiers	13
2.1.5 Active Contours	17
2.1.6 Template Matching	20
2.1.7 Geometrical Shape Models	20
2.1.8 Discussion	23
2.2 Problems addressed in this Work	25
2.2.1 Confocal Fluorescent Microscopy	25
2.2.2 Scope for New Algorithms	28
3 GMM Shape Modelling for Spot Detection	31
3.1 Introduction to the Biology	31
3.2 Introduction to Analysis	33
3.3 EM Algorithm for GMM Parameter Estimation	36
3.3.1 Likelihood	36
3.3.2 Hidden Variable	37

3.3.3	Parameter Estimation	38
3.3.4	Limitation of the EM Algorithm	39
3.4	The SMEM Algorithm	40
3.4.1	Basic Split and Merge	40
3.4.2	Modified Merge	41
3.4.3	Modified Split	41
3.4.4	Merge and Split Criteria	43
3.4.5	Discussion	44
3.5	The eSMEM Algorithm for Shape Modelling	46
3.5.1	EM Algorithm for Shape Modelling	46
3.5.2	Initialization of the GMM	48
3.5.3	Split and Merge Operation	49
3.5.4	Error Distribution	50
3.5.5	Error-based Split Method	51
3.5.6	Split and Merge Criteria for shape modelling	56
3.5.7	eSMEM Modeling Process	58
3.6	Conclusion	61
4	Spotnick: a practical eSMEM shape modelling tool	63
4.1	Pre-processing Stage	64
4.1.1	Noise Reduction	65
4.1.2	Spot Segmentation	65
4.1.3	Local Background Subtraction	71
4.2	Shape Modelling Results	72
4.2.1	Ground-truth Evaluation on Synthetic Samples	73
4.2.2	Comparison with the Zhang <i>et al.</i> SMEM algorithm [99]	83
4.3	Results on Real Data	91
4.4	Data Analysis	98
4.4.1	Parameter Registration and Statistics	102
4.4.2	Co-localization Analysis	102
4.5	Conclusion	106
5	Synapcut: a wavelet-based Bayesian framework for synapse segmentation	109
5.1	Introduction	110
5.1.1	Challenges in Synapse Segmentation	112
5.1.2	Scope of the Segmentation Model	112
5.2	Bayesian Segmentation Model	114
5.3	Likelihoods	116
5.3.1	Wavelet-likelihood	116

5.3.2	Global Intensity Likelihood	122
5.3.3	Inter-slice Intensity Likelihood	125
5.4	Smoothness Frame Prior	127
5.5	Graph-cuts Optimization	129
5.5.1	Graph-cut Model	132
5.5.2	Graph-cut Segmentation	132
5.6	Segmentation Results	134
5.7	Conclusion	136
6	Cellsake: an active contour model for cell/fibre segmentation	141
6.1	Biological Background	142
6.2	Active Contour	147
6.2.1	Numerical Solution	148
6.2.2	A Review of the Existing Conditional Forces $\mathbf{F}_{con}$	149
6.2.3	Discussion	151
6.3	Pre-processing	153
6.4	Cellsake	155
6.4.1	Environment Forces $\mathbf{F}_{env}$	156
6.4.2	Cellular Module $\mathbf{F}_{cell}$	160
6.4.3	Practical Modifications	166
6.4.4	Combining the Forces	170
6.4.5	Cellsake and the Merge scheme	172
6.5	Segmentation Results	173
6.6	Conclusion	177
7	Conclusion	185
7.1	Microscopic Image Segmentation	185
7.2	Spot Detection and Analysis	186
7.3	3-D Shape Reconstruction	187
7.4	Segmentation of Overlapped Objects	188
7.5	Future Work	190
A	Protein Synthesis and Regulation in Neuron Cells	193
B	Spotnick Graphical-User-Interface (GUI)	195
	Bibliography	203

List of Acronyms

AAM	Active Appearance Model
AC	Active Contour
ASM	Active Shape Model
DGMM	Dynamic Gaussian Mixture Model
DoG	Difference of Gaussian
DTCWT	Dual-Tree Complex Wavelet Transform
DWT	Discrete Wavelet Transform
eSMEM	error-based Split and Merge Expectation Maximization
EM	Expectation Maximization
GMM	Gaussian Mixture Model
GVF	Gradient Vector Flow
LoG	Laplacian of Gaussian
LSE	Least Square Error
LTM	Long Term Memory
mRNA	messenger Ribonucleic Acid
mRNP	messenger Ribonucleotide Protein
MAP	Maximum A Posteriori
MRF	Markov Random Fields
PDF	Probability Density Function
PSF	Point Spread Function

RNA Ribonucleic Acid

ROI Region of Interest

SMEM Split and Merge Expectation Maximization

SVM Support Vector Machine

VFC Vector Field Convolution

WT Wavelet Transform

1

Introduction

Since the first microscope was developed in 1590, microscopy has steadily improved and found increased use in biological research and clinical medicine as well as in many other fields. Microscopic imaging provides visualization of a broad range of biological and inorganic specimens at micro- to nano- scale levels, and enables researchers to investigate the detailed insights into the complex relationships that exist within the objects too small to be seen with the naked eye. Nowadays, the images observed by the microscopes are converted into digital data for easier storage, analysis or processing prior to display and interpretation.

As the amount of image data collected from various microscopic objects is ever-increasing, image processing has steadily become an indispensable part in microscopy because of the potential for large scale statistical evaluation on the acquired data. This is in addition to classical manual screening by eye. Among the different aspects of image processing, segmentation is a fundamental task for detecting and extracting the objects of interest from the observed images for further investigation. Considering the importance of segmentation as the first step in many image analysis applications, a great variety of segmentation algorithms have been proposed for a wide range of microscopic images. These algorithms are highly specific to the needs of various applications involved with different microscopic objects, such as molecules, nuclei, cells or other histological structures. The various image segmentation techniques can be categorized into the image driven approaches that are solely based on the image features, and the model driven approaches that incorporate certain patterns of the objects of interest.

The image driven approaches are usually simple and have higher robustness for the different applications with simple objects. Conversely, the model driven approaches tend to be more

elaborate and case specific, and usually incorporate with a-priori shape information to segment the specific complex image objects. The goals in this thesis are to extract the specific microscopic objects in the different applications. Since the image objects have different complex shapes and the segmentation in each application is based on a-priori shape information of the corresponding objects, a novel shape model is designed to segment the objects in each task by incorporating the corresponding shape information.

The development of these shape models is motivated by investigations on the neurological study of memory formation [46] and the muscle regeneration [40]. The work may be divided into three parts:

1. The extraction and colocalization analysis of protein particles
2. The 3-D shape reconstruction of neuron synapses
3. The segmentation of overlapped muscle fibres/cells

1.1 Spot Analysis

The brain creates long term memory by synthesising certain proteins at stimulated synapses. Different types of proteins are involved in the synthesis process. The study of memory formation in [46] investigates the characteristics of the different proteins by considering the distribution of each type of protein in the neuron synapses and the colocalization of the different types of proteins. The different proteins in the living specimen are stained with different fluorescent markers and are imaged separately using a confocal microscope. A gray-scale image is generated for each type of protein, and the images of the different markers from the same specimen are registered to each other. The image processing task is to detect and measure the protein particles in each image and then analyze the relative separation of particles from each other across the two images (colocalization analysis). The difficulty of this task is that some protein particles are clustered and cause large and irregular spot appearance in the resultant image. In order to detect the ‘merged’ particles and estimate the overlapped shape for the further colocalization analysis, a Gaussian mixture model (GMM) is used to fit the intensity profile of each cluster spot and identify the individual particles by the components of the mixture model. Also, since the shapes of the particles are parameterized by the GMMs, the model parameters are used directly in the colocalization analysis.

1.2 3-D Shape Reconstruction

The shape of the neuron synapses is a valuable piece of information in the study of memory formation. The shape information of the synapses is captured by scanning the synaptic object at multiple different ‘depths’ from top to bottom of the living specimen, and is stored as an image

stack of the serially scanned images. The goal is to reconstruct the 3-D shape of the synapses from the observed image stack generated by a confocal microscope. This is a challenging problem because of the irregular luminance of the synapses and the presence of features that are out of focus. These are caused by the defect in fluorescent staining process and the physical limit of the confocal microscope. A Bayesian model is introduced for the segmentation which incorporates the ‘sharpness’ information of the objects, the global intensity histogram, and the inter-slice intensity information. The ‘sharpness’ information is assumed to be luminance invariant and is derived by the high frequency content of the image in the wavelet domain. The global intensity histogram is used to separate the potential object from the background noise. The inter-slice intensity information is the intensity profile across the slices of the image stack at the same pixel site of every slice. The sharpness and the intensity information give rise to three different likelihood terms in the model. Also, the model employs a prior that is constrained by the ‘frame’ of the synapse. The frame can be considered as the skeleton of the winding synapses. Since the synapses are connected in some patterns, the idea of the frame prior is to enhance the connections of the synapses which are not considered in the likelihood terms. Finally, the Bayesian segmentation is optimized using the graph-cut method.

1.3 Segmentation of Overlapped Objects

A novel muscle regeneration technique that uses electrical stimulation to control the growth of the muscles is under investigation for skeletal muscle therapy [40]. The development of the muscle sample is monitored via a confocal microscope. The cells and fibres (formed by multiple cells) need to be extracted from the observed images in order to validate and evaluate the performance of the new technique. Hence, the task is to segment and classify the cell and fibre objects in the observed images. The classification of whether an object in image is a fibre or a cell is based on the number of nuclei inside the cell body of that object. The nucleus detection is similar to the spot analysis in the first task, and the goal in this task is to estimate the shape of the cell body objects so that the shape estimation of each corresponding object covers the correct nuclei belonging to that object. Since the images are 2-D projections of the 3-D test samples, the actual cells/fibres distributed at different depth levels in an observed sample may be projected as overlapped objects in the resulting image. Since both the fibres and the overlapped objects have complex shape appearance in the image, the overlapped objects can ‘confuse’ the classification of individual fibres and hence the segmentation of the cell body objects becomes challenge. Since there is no information for the occluded parts in the overlap regions, the estimation relies on the non-overlapped parts of the objects that inform appropriate shapes for occluded parts. Considering the simpler structure of the single cells in comparison with the complicated fibre shapes, and the fact that each fibre is formed by combining multiple cells, the idea of the proposed algorithm is to divide each separate region of interest (ROI) in the cell-body image into small ‘cell candidates’, and fitting an AC to the corresponding cell candidate in the

ROI. The key is the development of specifically purposed novel forces to constrain the growth of the ACs and recover the occluded parts of the candidates in the overlap region. Finally, a merge scheme is developed for determining and merging the ACs that belong to the same fibre. The shapes of the cells/fibres in the overlapped object are then estimated by the resultant merged shape of the ACs.

1.4 Thesis Outline

A brief summary of each chapter is as follows.

Chapter 2 Image Segmentation in Microscopy: A Review

This chapter gives a brief overview of various image segmentation techniques proposed in the previous work in microscopy. The techniques are categorized as either image driven approaches or model driven approaches. The image driven approaches include image thresholding, watershed segmentation and data clustering. The model based approaches consist of template matching, classification models, active contours and geometric shape models. This is followed by a discussion about the scope of the algorithms introduced in this work and the motivation from the previous work. Also, a brief description of the confocal fluorescent microscopy is presented to show how the images concerned in this work are generated.

Chapter 3 GMM Shape modelling for spot detection modelling

This chapter provides a brief introduction to the biological background of the memory formation, and outlines the novel application of GMMs for fitting the intensity profile of spots in a given image in order to detect protein particles. Since the GMMs are optimized using a split-and-merge expectation maximization (SMEM) algorithm, a brief review and discussion of the previous SMEM algorithms are presented. This is followed by the introduction of a novel error-based SMEM (eSMEM) algorithm that is customized for the GMMs in shape modelling.

Chapter 4 Spotnick: a practical eSMEM shape modelling tool

This chapter shows how the eSMEM algorithm introduced in *Chapter 4*, can be employed to perform spot modelling and analysis on real biological data. The key addition is the introduction of the pre-processing stage to overcome the scanning-noise and the background luminance problems. The complete shape modelling algorithm (named ‘Spotnick’), including both the pre-processing and the eSMEM algorithm, is compared with the Zhang *et al.* SMEM algorithm [99] on a large amount of synthetic mixture samples with different numbers of components. The real data results of Spotnick are presented and show appropriate estimates to the ‘merged’ spots.

Chapter 5 Synapcut: a wavelet-based Bayesian framework for synapses segmentation

This chapter presents the novel Bayesian model for segmenting 3D synaptic objects from a given image stack. A detailed description of the likelihood terms and the shape prior is presented. Summaries of the segmentation results and a discussion of the future improvement are also presented.

Chapter 6 Cellsnake: an active contour model for cell/fibre segmentation

This chapter outlines the new active contour mode (Cellsnake) for shape estimation of the overlapped cell/fibre objects in order to detect the objects from the combined appearance in the microscope images. The chapter details the energy forces of the model derived from the a priori shape information are presented. The segmentation results the synthetic data and real data are presented and are discussed in the final section.

Chapter 7 Conclusions

This final chapter assesses the contributions of the research and outlines some possible improvements and idea for future work.

1.5 Publications

Portions of the work described in this thesis have appeared in the following publications:

- [78] K. Pan, J. Hillebrand, M. Ramaswami, and A. Kokaram. Gaussian mixtures for intensity modelling of spots in microscopy. In *International Symposium on Biomedical Imaging (ISBI'10)*, pages 121 - 124, Rotterdam, Netherlands, February 2010.
- [77] K. Pan, J. Hillebrand, M. Ramaswami, and A. Kokaram. Gaussian mixture models for spots in microscopy using a new split/merge em algorithm. In *IEEE International Conference on Image Processing (ICIP'10)*, pages 3645 - 3648, Hong Kong, China, October 2010.
- [46] J. Hillebrand, K. Pan, A. Kokaram, S. Barbee, R. Parker, and M. Ramaswami. The me31b dead-box helicase localizes to postsynaptic foci and regulates expression of a camkii reporter mrna in dendrites of drosophila olfactory projection neurons. In *Frontiers in Neural Circuits*, 4(00121), 2010.
- [79] K. Pan, A. C. Kokaram, K. Gilmore, M. J. Higgins, R. Kapsa, and G. G. Wallace. Cellsnake: A new active contour technique for cell/fibre segmentation. In *IEEE International Conference on Image Processing (ICIP'11)*, pages 2153 - 2156, Brussels, Belgium, 2011.

-
- [76] K. Pan, D. Corrigan, J. Hillebrand, M. Ramaswami, and A. Kokaram. A wavelet-based Bayesian framework for 3d object segmentation in microscopy. In *SPIE BiOS Symposium*, San Francisco, United States, 2012.

2

Image Segmentation in Microscopy: A Review

Microscopy is the technical field of using microscopes to view samples and objects that cannot be seen with the naked eye. Microscopic imaging enables the scientists to observe cells and molecular dynamics within the living cell at micro- to nano- scales, and allow them to acquire static images of the structures of interest in two or three dimensions in a living specimen and can also record time-lapse image sequences to study molecular transport or transformation within the cell and in the extracellular substance. As the imaging techniques in microscopy are ever-improving, image processing, as an integral part of microscopy, is increasingly important to the fields of medicine and scientific research for analysing the observed data for further image understanding. One of the popular strategies in microscopic image processing is image segmentation for extracting the objects of interest (i.e. molecules, cellular subunits, whole cells, tissues or organs). This task is made difficult because of the adverse environment in which the pictures are acquired. Typically images are corrupted by noise, blur and uneven contrast. Hence many image segmentation techniques developed for microscopy applications tend to focus on particular objects of interest. This chapter presents a brief overview of the existing techniques for microscopic image segmentation, and also includes a discussion of the tasks addressed in this work and the outlined shape analysis solution motivated by the previous techniques.

2.1 A Review of Segmentation Techniques in Microscopy

Microscopic imaging provides visualization of a wide range of biological and inorganic specimens at micro- to nano- scale levels, and enables researchers to investigate the detailed insights

into the complex relationships that exist within living cells. However, with an ever-increasing amount of multi-dimensional image data of various microscopic objects in biomedical research and clinical medicine, quantitative interpretation of the data becomes complex, time-consuming and possible subjective due to different visual interpretations of different researchers. Consequently, microscopic image processing is an indispensable part of microscopy which allows large scale statistical evaluation on the acquired image data in addition to classical eye screening qualitative valuation, and extracts the valuable information for further biological analysis.

Image segmentation is the fundamental process for the analyses in microscopic images which provide important information for further image understanding by extracting the sub-cellular particles, nuclei, cells or histological structures from stained tissues with different markers with an adequate accuracy. However, the segmentation process is challenging due to not only the large variations of the features (size, shape, orientation or texture) in the desired objects in different applications, but also the co-existence of unwanted objects, and variations in background luminance. In many image analysis applications, segmentation is often the first and the most difficult step. Due to its importance, a great variety of segmentation algorithms have been proposed for a wide range of applications. Also, the publications are wide-spread in literature: microscopy, biomedical engineering, biomedical imaging, bioinformatics and pattern recognition (some of them are reviewed in this section).

The task of segmentation in microscopic images refers to the process of finding the regions (boundaries and area) of the desired objects with an adequate accuracy from the images of stained specimens with different markers. The considered papers study different image types from the prostate, breast, blood, bone marrow, cervix, esophagus and lymphatic-system obtained with different markers, e.g., DAPI, immunodetection of lamin A/C, hematoxylin and eosin, Papanicolaou stain.

In general, the various segmentation algorithms can be categorized as image driven or model driven. Image driven approaches operate directly on the image pixels and detects objects solely based on the image features. This includes image thresholding, watershed segmentation and data clustering. Conversely, model based approaches assume that objects in an image are present in a certain pattern. A list of models generally used for image segmentation are template matching, classifications models, active contours and geometric shape models.

The following sections present a brief overview of the existing techniques in different applications and the publications in a categorized order. Because almost all proposed techniques combine at least two image processing methods, the techniques are grouped based on the relevant/critical image processing method which distinguishes them from other approaches.

2.1.1 Thresholding

Thresholding is a simple method for segmentation and the key is the choice of the threshold value. There are three popular methods for image thresholding in microscopy, which are Otsu's method

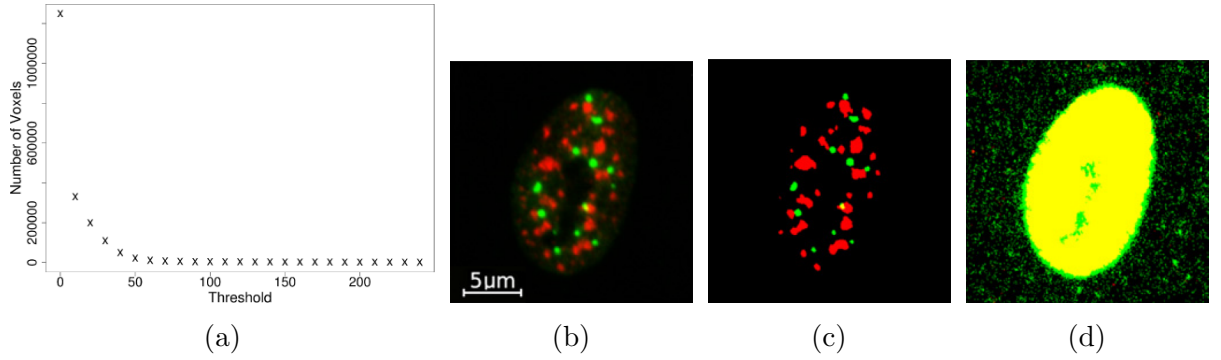


Figure 2.1: The idea and the result of the SCT algorithm (a) plot of the number of voxels with an intensity greater than or equal to the given threshold in a typical image stack of an MRC5 cell nucleus with fluorescently tagged PML; (b) an image of nuclei in MRC5 primary fibroblasts ('red' for SC35 protein, 'green' PML protein in); (c) segmentation produced by applying the SCT algorithm to each colour channel of (b); (d) segmentations produced by applying Otsu's algorithm to each colour channel of (b). (Reproduced from [84])

[75], Ridler and Calvard's iterative-selection method [83] and Kittler and Illingworth's minimum-error-thresholding method [55]. To find the optimum threshold, Otsu suggested minimizing the weighted sum of within-class variances of the foreground and background pixels, which is equivalent to maximizing the scatter between the classes. Ridler and Calvard's iterative-selection method is an iterative version of Otsu's method. In each iteration, a new threshold is established using the average of the foreground and background class means. The iterations terminate when the change in the repeatedly updated threshold becomes sufficiently small. In Kittler and Illingworth's minimum-error-thresholding method, each class of the pixels is characterized by a conditional Gaussian probability density functions, and the optimum threshold is yielded by minimizing the total mis-classification error.

Microscopy images are rarely so simple and noise-free that a single threshold would be robust. Hence a number of adaptive thresholding schemes have been developed. In [59], a multi-thresholding scheme was employed to extract sub-cellular structures by assuming that there were three different classes of features in the image. Barrera et al [7] performed spot detection by applying locally adaptive thresholding to cope with irregular illumination across the image. Russel et al [84] segmented sub-cellular structures by a simpler adaptive thresholding technique. In this technique the threshold is increased until the area of detected objects reaches a steady state (see Fig. 2.1).

2.1.2 Watershed Segmentation

Conventional thresholding methods fail because they do not take into account either the spatial correlation of the objects of interest, or the structure of those objects. The watershed segmentation algorithm has been used successfully to exploit spatial information implicitly. The result is

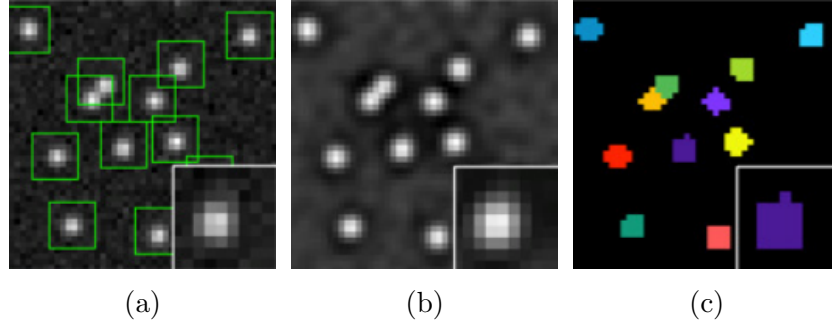


Figure 2.2: Illustration of the Wavelet-based segmentation in [49] (a) *Source images with the localization of single molecules in green areas.* (b) *The lo-lo band at level 2 of the wavelet transform.* (c) *The results of wavelet segmentation.* (Reproduced from [49])

better separation of objects which share a boundary. Typically, the watershed algorithm is used to segment a relief function related to the image (typically gradient magnitude). The watershed works by iteratively ‘flooding’ the relief function (started from the local minima) and identifies the segments by keeping track of the different connected regions.

Izeddin et al [1] improved the performance of watershed segmentation for this problem by applying it to the 2nd level wavelet decomposition of the image (see Fig. 2.2).

However, the direct application of the basic watershed algorithm generally leads to over-segmentation due to noise and other local irregularities of the gradient. There are three approaches capable of reducing this problem. The first option is to pre-process the image. The second one is to merge adjacent regions according to some criteria after the use of watershed. The last approach is the marker-controlled watershed algorithm where the local minima (the markers) are firstly eliminated based on particular criteria before the ‘flooding’.

Cheng and Rajapakse [24] proposed a marker-controlled watershed algorithm for separating the clustered nuclei in fluorescence microscopy cellular images (see Fig. 2.3), where the undesired regional minima are removed by using an adaptive H-minima transform. In [24], the regions of clustered nuclei are firstly detected by using an active contour model. A set markers for separating the clustered nuclei are initialized by the minima of the inverse inner distance map of each of the region. The adaptive H-minima technique uses the original H-minima transform [88] to eliminate the unwanted markers.

In [51], Jeong *et al.* proposed a hybrid of the marker-controlled and region merging approaches. In this method, an technique using ultimate erosion (iteratively erode a figure to get a skeleton) is proposed for detecting the markers for the clustered nuclei. Also, Jeong *et al.* introduced a stepwise merging strategy to merge the over-segmented fragments from the watershed algorithm into an isolated nuclei (see Fig. 2.4).

Moreover, the markers can also be extracted by using the morphological reconstruction technique proposed by Vincent in [90]. This technique is essentially a generalization of flood-filling. It processes one image (the marker) based on the characteristics (e.g. the gradient) of

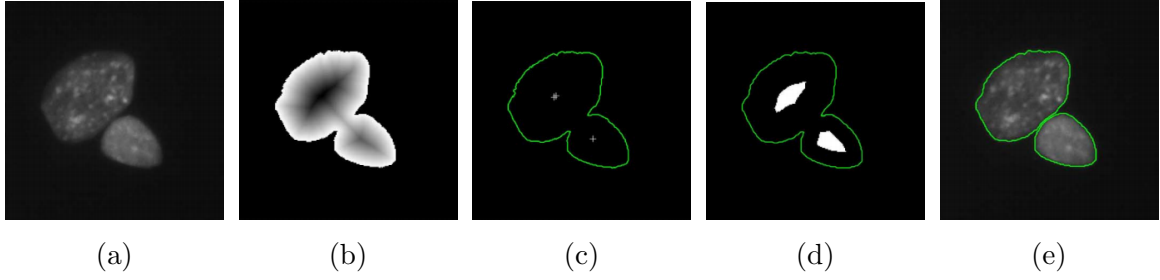


Figure 2.3: Illustration of the marker-controlled watershed algorithm in [24] (a) An example of the clustered nuclei. (b) The inverse inner distance map generated after segmentation of the cluster shape. (c) the grey spots show the local minima of (b). (d) The shape markers in white obtained from adaptive H -minima transform. (e) The results of the watershed segmentation. (Reproduced from [24])

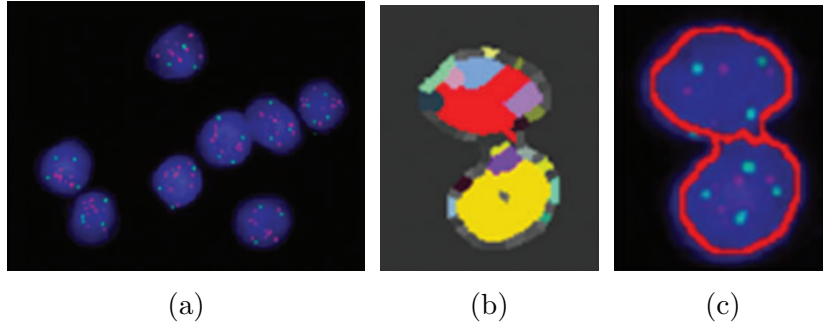


Figure 2.4: Illustration of the nuclei classification based watershed segmentation in [51] (a) Image of leucocyte nuclei. (b) Over-segmentation after watershed segmentation. (c) Final results after stepwise-merging strategy. (Reproduced from [51])

another image (the mask). The algorithm begins from the peaks in the marker image, where the peaks are forced to spread out, or dilate to fit within the mask image. The spreading process continues until the image values stop changing. The desired markers for the watershed approach are assigned by the regional maxima extracted by subtracting the marker image from the mask image. Since the morphological reconstruction technique extract the regional maxima rather than the local maxima, the unwanted markers are hence eliminated.

2.1.3 Data Clustering

Data clustering methods are widely used in image segmentation. The idea is to map each pixel or patch of pixels into some feature space and then cluster the feature vectors in that space. Each cluster so discovered is then assumed to represent a region or segment in the image space. The image pixels or patches are then segmented into different regions through their cluster membership. The most popular methods in data clustering are k-means clustering, fuzzy c-means, mean-shift clustering and Gaussian mixture model (GMM).

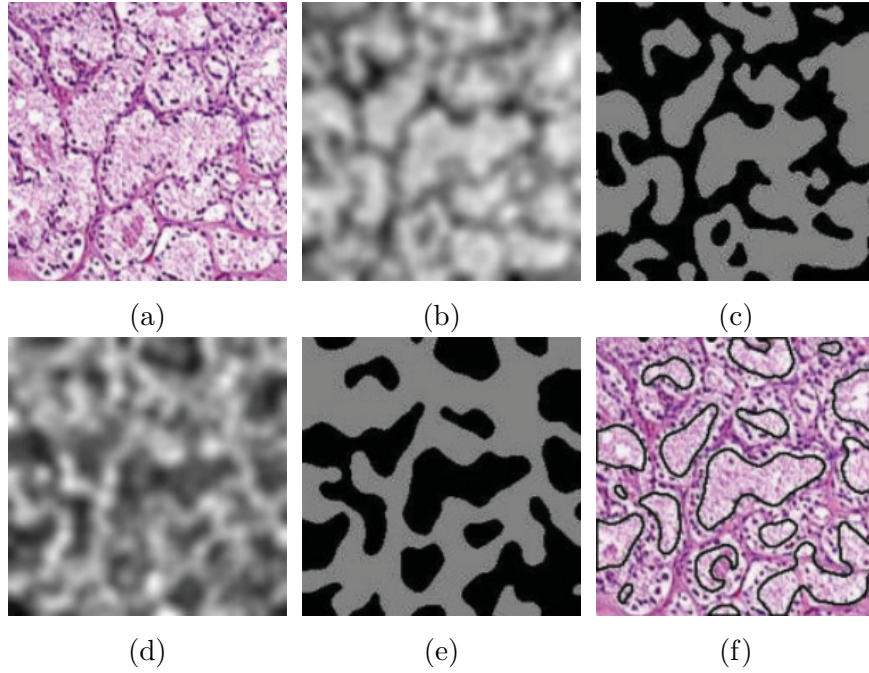


Figure 2.5: Illustration of the texture-based k-means clustering segmentation in [35] (a) *RGB image of the glands.* (b) *Feature map of the gray-scale version of (a).* (c) *Segmented image of (b) by using k-means clustering.* (d) *Feature map of the variance density image of (a).* (e) *Segmented image of (d) by using k-means clustering.* (f) *The boundaries are found by combining (c) and (e). The original image (a) overlaid with the estimated segmentation boundaries* (Reproduced from [35])

K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results in a partitioning of the data space into Voronoi regions. In [35], Farjam *et al.* segmented the gland components (stroma, lumen, nuclei) in the prostate tissue images by grouping the roughness texture features for each object using the k-means clustering (see Fig. 2.5).

Fuzzy c-means clustering allows each point to have a degree of belonging to clusters, as in fuzzy logic, rather than membership completely to just one cluster. Thus, points on the edge of a cluster, may be in the cluster to a lesser degree than points in the centre of cluster. In [26], a cluster analysis based on the fuzzy c-means algorithm is performed by Colantonio *et al.* on the HSV colour feature of the original RGB image of lymphatic tumours to provide a rough automatic segmentation of the clustered cells (see Fig. 2.6).

Mean-shift clustering is a non-parametric clustering technique which does not require prior knowledge of the number of clusters, and does not constrain the shape of the clusters. The mean shift is a simple gradient technique used to find the modes of the kernel density estimate of the clusters, where the modes are equivalent to the location of the densest area of the data set where these locations could be satisfactory cluster centre estimates. In [82], Qi

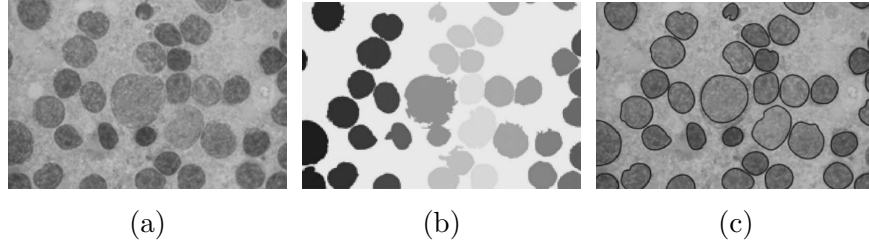


Figure 2.6: Illustration of the segmentation using fuzzy clustering and neural discrimination in [26] (a) *The image of the lymphatic tumours.* (b) *Labelling of the connected regions identified during fuzzy clustering.* (c) *Identified contours of each cell based on (b).* (Reproduced from [26])

et al. integrated the mean-shift algorithm with a single-pass voting approach to automatically locate the geometric centres of the overlapped cells in breast tissue images.

Gaussian mixture models are among the most statistically mature methods for clustering, where the distribution of each cluster is fit to a mixture of Gaussian densities. The parameters of the GMMs are usually optimized using the expectation-maximization (EM) algorithm. This is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models and is proposed by Dempster in [32]. In [8], Begelman *et al.* proposed a prostate tissue segmentation algorithm that uses a GMM to group the colour features of different objects (stroma, nuclei, and prostate glands) by fitting the GMM to the colour distribution. Each Gaussian in the GMM is assumed to represent one object type, and the GMM is optimized by the EM algorithm. Usually, the GMM is an image driven technique in the sense that it operates in a feature space derived from the image. However it can be used instead as a model for shape fitting in which case the GMM surface is used as the function fit for some observed shape or volume. The dynamic Gaussian mixture model (DGMM) introduced by Xiong *et al.* in [92] is a good example. The DGMM is proposed for cell tracking in time-lapse image sequence (Fig. 2.7(a)&(b)). Each cell image is firstly thresholded in order to extract the feature points in the cells, and then a GMM is fitted to the clusters of the feature points (Fig. 2.7(c)) so that each component of the GMM informs the position and shape of the corresponding cell. The modelling of the cellular shape is actually GMM clustering of the feature points (extracted by thresholding) of the objects (Fig. 2.7(c)). The optimization of the model parameters is implemented with the EM algorithm.

2.1.4 Machine Learning Classifiers

Ultimately, image segmentation is achieved by classifying pixels according to the probability that they belong to certain image classes. Hence many classifiers can be used in this application. The popular frameworks used previously in microscope image processing include SVMs (Support Vector Machines), Neural Networks and general Bayesian frameworks for classification.

Both SVMs and Neural networks have very long and rich histories in the literature of Machine

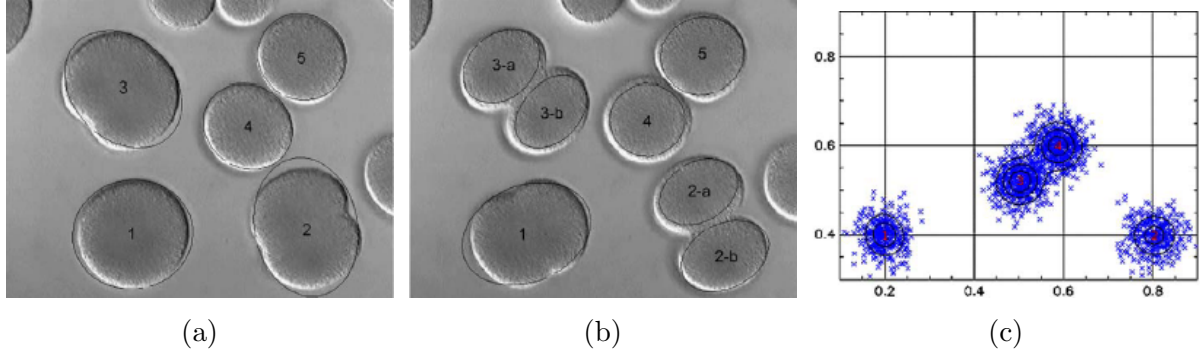


Figure 2.7: Gaussian mixture modelling for cell tracking in time-lapse image sequence ((a) and (b) are two adjacent frames in a time-lapse image sequence. The spots labelled with 2 and 3 in the (a) are split into the spots 2a&2b and 3a&3b respectively in the next frame (b) of an image sequence. (c) shows that the GMM is estimated from the elliptical clusters of the feature points after thresholding a given cell image.) (Reproduced from [92])

Learning. The reader is directed to excellent texts by Duda *et al.* [34] and Murphy [70] that describe these techniques in detail. For the purposes of this thesis it is sufficient to note that the SVM in its simplest form is a linear binary classifier while Neural networks are used for non-linear multiclass classification. In both cases the idea is to define a discrimination boundary between classes in the relevant feature space. The first step in classification is typically to estimate the parameters of the classifier using labelled training data. Segmentation is achieved by estimation of the class membership at each pixel or patch site.

A hybrid colour SVM is proposed by Lebrun *et al.* in [57]. Given a multi-channel image with more than two classes of different objects, each class is associated with a SVM classifier. By subjecting a colour feature to the SVMs, the output of each SVM for every pixel is mapped to a posterior probability by applying a sigmoid function. Then, the pixel probability image for each class (Fig. 2.8(d)-(f) for three different classes) is generated for the segmentation using watershed algorithm, and the segmentation result is shown in Fig. 2.8(c).

In addition, the SVM is often used to identify the phases or grades of the detected objects. For instance, the cell cycle phase identification system proposed by Wang *et al.* in [91] uses an online updating SVM to identify the cell cycle phase of the cell detected by watershed segmentation. Also, the morphological features of the gland objects detected by the active contour analysis proposed by Naik *et al.* in [72] are subjected to a SVM algorithm to identify the intermediate Gleason grades of the given prostate histology image.

In [11], a neural network classifier is applied by Boland and Murphy to determine the localization patterns of the sub-cellular organelles or structures in order to find the context of the protein's function. The sub-cellular location features extracted from the target objects are subjected to a back-propagation neural network trained by a fixed number of instances from each class, and hence identify the localization pattern. Similarly, an error back-propagation neural

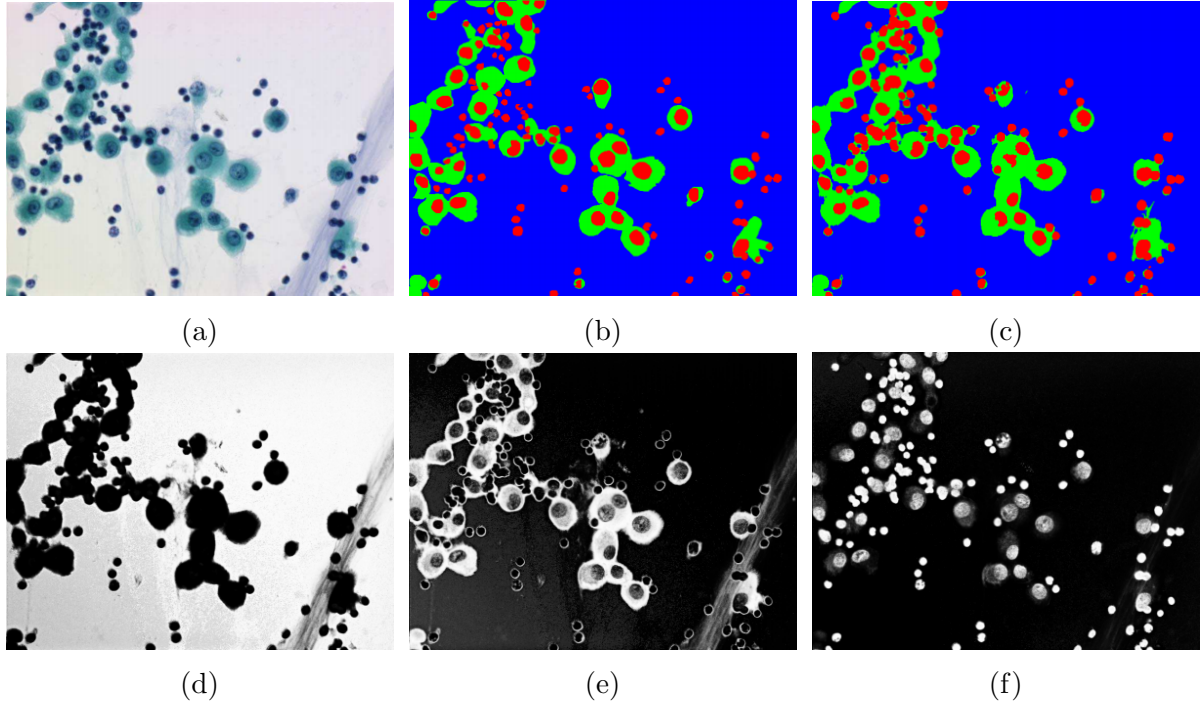


Figure 2.8: Illustration of the SVM probability based watershed segmentation in [57] (a) *Original image*. (b) *Expert segmentation: background (blue), cytoplasm (green), nucleus (red)*. (c) *Results of watershed segmentation*. (d) *Pixel probabilities estimation of background* (d), *cytoplasm* (e) and *nucleus* (f). (Reproduced from [57])

network is proposed by Colantonio *et al.* in [26] to identify the classes of touching objects in lymphatic tumours images in order to isolating and define the boundaries of the objects.

Bayesian segmentation is a statistically based classification method, and is a convenient way of taking a priori information into consideration when inference is made from images. Assuming that there are say 2 segments to be estimated in the image, the problem is posed as that of estimating a segmentation label at each pixel site. That label takes on one of two values in this case say 0,1. In a Bayesian fashion, the problem is to maximise $p(L|I)$ over the image space I and that is written $p(L|I) \propto p(I|L)p(L)$. Prior information typically about spatial smoothness is injected as $p(L)$ and the image appearance given each label L is modelled by defining $p(I|L)$, the likelihood. See [36] for more details about Bayesian segmentation.

In [71], a Bayesian classifier is applied by Naik *et al.* to extract the low-level information of the target objects for the gland and nuclei segmentation. The Bayesian classifier estimates the likelihood of each pixel belonging to an object of interest (lumen, cytoplasm or nuclei) based on the RGB intensity. The likelihood image for each class of object is then used to detect the nuclei and to find the boundary of the gland objects.

Bayesian segmentation is most powerful when spatial priors are used in the inference. The most popular such prior is the Markov Random Field (MRF). Label estimation is performed us-

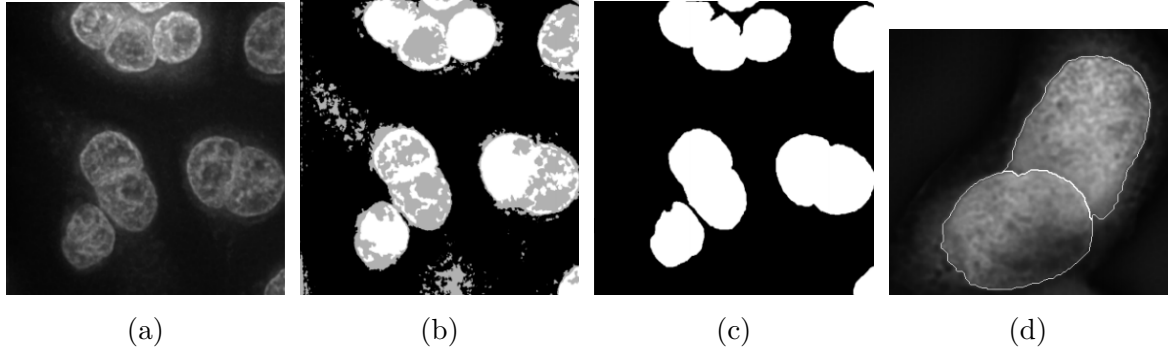


Figure 2.9: Illustration of two-stage graph-cut segmentation model in [31] (a) *Original image.* (b) *Foreground (white) and background (black) markers.* (c) *Background segmentation.* (d) *Separation of the isolated cells in the clusters* (Reproduced from [31])

ing any number of optimization methods. The most powerful of these is the graph-cut technique first proposed in [42]. That technique is guaranteed to yield the optimal solution in the case of two class labelling but a number of extensions have been developed for multi-class problems since then e.g. [15]. In [31], a two-stage graph-cut model is introduced by Daněš *et al.* to segment the clustered cell nuclei (see Fig. 2.9). The first stage of the model is a background segmentation (Fig. 2.9(c)). Voxels with very high probability of being foreground or background are located and separated by a boundary with a minimal geodesic length. Also, a bi-level histogram analysis is applied to define the background/foreground markers (Fig. 2.9(b)) when the intensities of the voxels are below the background threshold or above the foreground threshold. The second stage is to split the isolated cells from the obtained clusters by combining image gradient information and a priori knowledge about the shape of the nuclei. The voxels are associated with a probability value derived from the Euclidean distance of the voxels to the nearest peak, and are separated by the minimum cut of the graph-cut model (Fig. 2.9(d)).

Similarly, a (MAP-MRF) energy function minimization problem is also posed in the automated 3D reconstruction framework introduced by Yang and Choe in [95] to segment the neural cells from 3D anatomical brain models. The idea of the framework is to segment the neural regions in each 2D image first and then to find the region correspondences between adjacent images. The segmentation is posed as a (MAP-MRF) energy function minimization problem. The energy function to be minimized contains a regional term and a boundary term. The regional term is defined over the image gradient vector fields the shape information which is formulated as the distance function. On the other hand, the boundary term is defined over the gray-scale intensity of the image. The energy function incorporating the two terms is minimized using the graph-cuts algorithm.

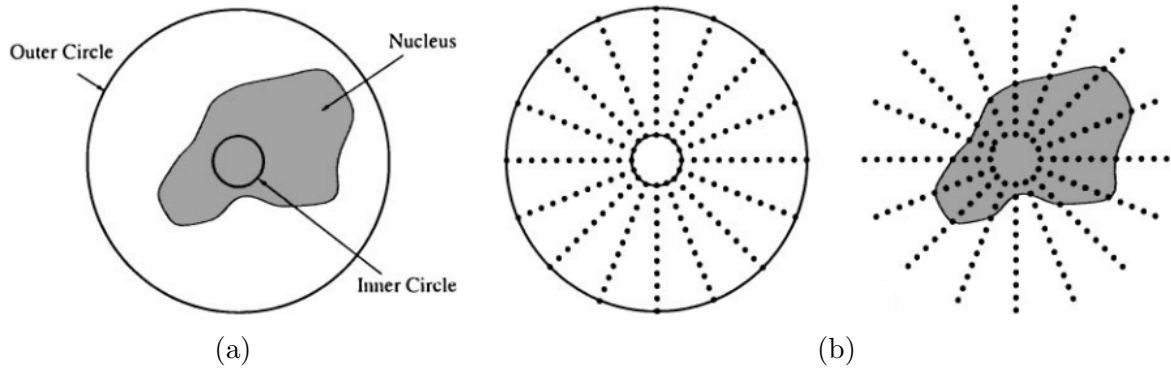


Figure 2.10: Illustration of the Viterbi search-based dual active contour model in [4] (a) *Cell nucleus with inner and outer contours (circles)*. (b) *Discretization of the search space*. (Reproduced from [4])

2.1.5 Active Contours

Active contours (ACs) are shape-based model driven approaches which are considered as the state-of-the-art in medical image segmentation and have recently been used for the segmentation of biological images. Unlike all of the methods examined so far, Active Contours explicitly model the shape contour, rather than implicitly as the boundary of some segmentation of the image. An AC (also called snake) is a deformable spline influenced by constraint and image forces that pull it towards object boundaries. The principle of ACs is to search for a curve in the image where the weighted sum of internal (constraint) energy and potential energy (image force) is minimum. The internal energy is defined by the curve itself in such a way as to keep the contour smooth during deformation. The potential energy is computed from the image data to move the curve toward an object boundary or other desired feature within an image.

In [4], a Viterbi search-based dual active contour model is introduced by Bamford and Lovell for cell nucleus segmentation in stained cervical cell images. The approach of the model is to use inner and outer contours (Fig. 2.10(a)) which are required to lie within and outside the object of interest, respectively, as opposed to near the final solution. A driving force pushes the contours towards each other which enables each contour to overcome local minima in the image. The process is halted when both contours meet. The globally minimum energy contour is firstly defined by using the Viterbi algorithm in the discrete search space (Fig. 2.10(b)) between the inner and outer contours. The final smooth boundary is found by attracting the ACs to the defined globally minimum energy contour.

An improved active contour model is proposed by Hu *et al.* in [48] to isolate each cell nucleus in microscopic images of esophageal smears. Firstly, the cell nuclei are detected by using morphological and operators. After the detection, an improved active contour model is built for each nucleus to track its boundary. The model combines the traditional snake with a growing energy. This 'growing' energy is based on sub-block intensity similarity which expands

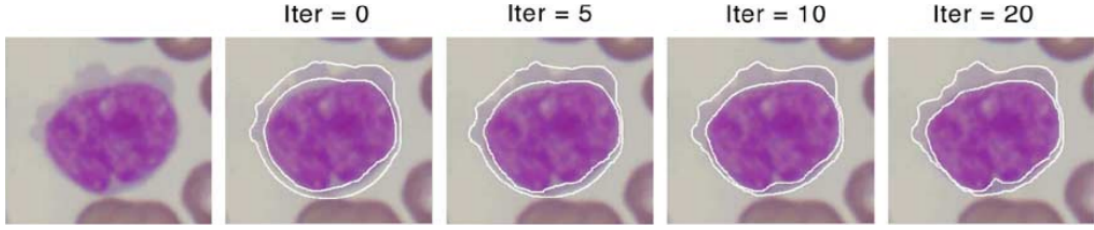


Figure 2.11: Segmentation results for the cytoplasm and nucleus by using the colour GVF active contour model (Reproduced from [96])

the contour outward from the nucleus centre detected on initialization.

A colour gradient vector flow (GVF) active contour model is proposed by Yang *et al.* in [96] for cell segmentation in colour images. The given RGB image is firstly converted to the LUV colour space which is more perceptually uniform for the Euclidean metrics and distances. The colour GVF is generated by the changing rates in the directions of the eigenvectors of the matrices created by the 2^{nd} derivatives of the L, U, V channels. In order to obtain a robust estimation of the initial locations of the nucleus and cytoplasm of cells without being affected by outliers (the objects without interest), a statistically robust matching criteria is incorporated with the active contour model to classify the objects before the evolution process of the snake. This classification is based on a minimum distance estimator that minimizes the integral squared error (L_2E) between a Gaussian model of the residual and the true density function of the residual. The segmentation results are shown in Fig. 2.11.

In [68], Mosaliganti *et al.* introduced a variational region-based active contour for density-based segmentation of multicellular structure (Fig. 2.12(a)). This model uses a level-set method based on the Mumford-Shah energy functional but replaces the scalar constants that approximate the foreground and background intensities with distinct and constant cellular densities. The cellular densities (Fig. 2.12(b)) are estimated using 2-point correlation functions computed on classified cell images. The segmentation results of the level-set formulation are given in Fig. 2.12(c).

On the other hand, some AC algorithms use reference images to incorporate the shape information of the target objects in constraining the AC deformation. An AC model using reference shape prior is introduced by Foulonneau *et al.* in [38]. In this AC model, the shape of reference object is firstly encoded by a shape descriptor defined by the Legendre moments derived from the inner region of the object. The shape descriptor is then subjected to a density function of mixture of Gaussians and gives rise to a shape prior energy. The evolution equation for the AC is derived from the minimization of the shape prior energy using a shape derivative framework. The segmentation results of objects with occlusions are shown in Fig. 2.13.

The idea of using reference object shapes is also adopted by two other popular active models, the active shape model (ASM) and the active appearance model (AAM) developed by Cootes *et al.* in [29,30]. The ASM is a statistical model of the shape of a set of reference objects which

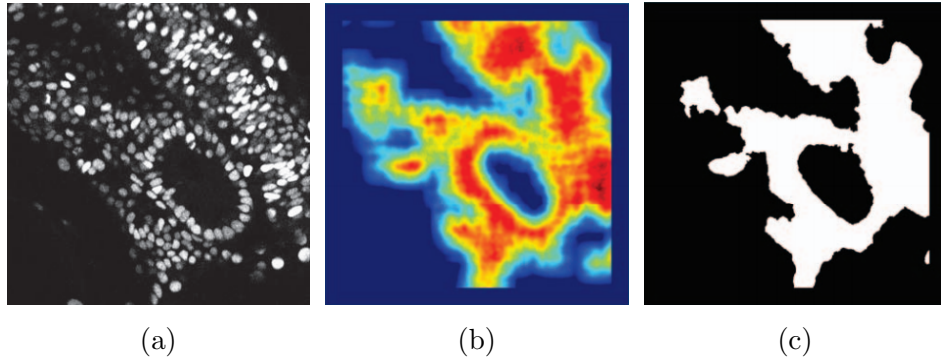


Figure 2.12: Segmentation results of the density-based level-set formulation in [68] (a) *A optical section of the zebrafish embryo ear. Note the variable density of cells.* (b) *Colour-coded density map of cell regions.* (c) *Two-class segmentation of the image using the density-based level-set formulation.* (Reproduced from [68])

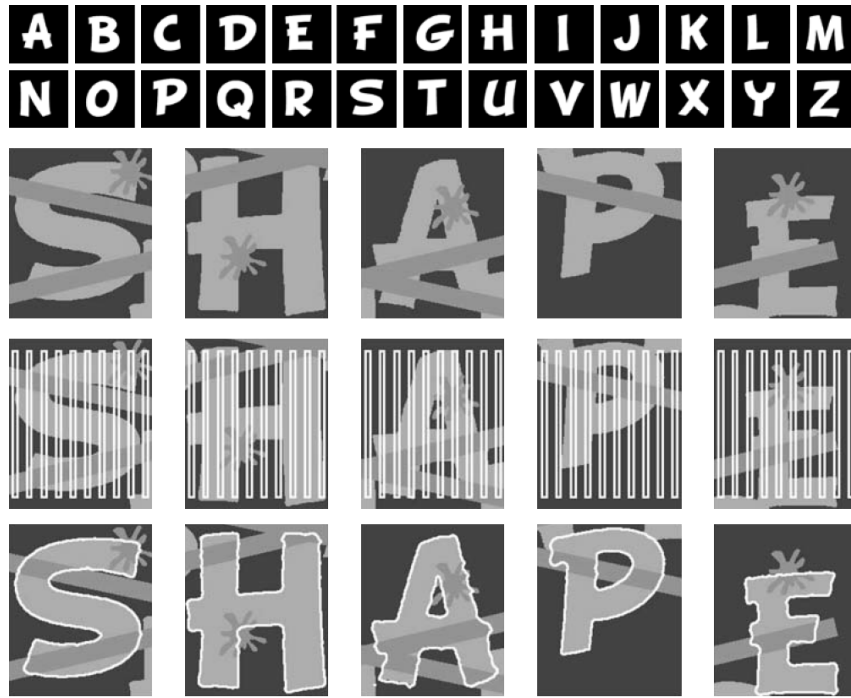


Figure 2.13: Segmentation results for the letters featuring large occlusions. *First two row: a set of reference shapes used with the reference AC model. Third row: original images with letters featuring occlusions. Forth row: initialization. Fifth row: final result with the shape constraint.* (Reproduced from [96])

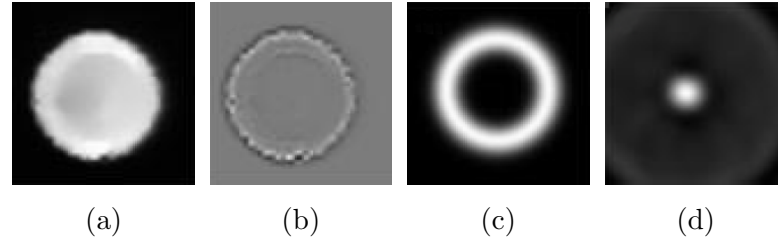


Figure 2.14: The spot identification process. (a) A vignette image containing a spot. (b) The Laplacian image of (a). (c) Template of desired boundary. (d) The 2D cross-correlation of (b) and (c). (Reproduced from [19])

iteratively deform to fit to an example of the object in a new image. The shape of ASM is represented by a set of points and is constrained by the point distribution model when fitting to the target object. The shape of ASM varies only in ways seen in a training set of labelled examples. AAMs are developed based on the ASMs by incorporating the appearance constraint derived from the texture across the target object. The AMM uses all the information in the image region covered by the target object, rather than just that near modelled edges, and the model parameters are estimated by minimizing the difference between the synthesised model and the image.

2.1.6 Template Matching

Template matching is a simple and effective method when the target objects have uniform shape appearance in the image. The template can be a fixed model that estimated from the appearance of the objects or is calculated from a number of manually selected training objects. The estimated object template is matched against the given image using normalized cross-correlation, and the positions of the objects are then identified by the local maxima (above a particular threshold value) in the resultant cross-correlation image.

In [19], a spot finding system based on template matching is proposed by Buhler *et al.* for the DNA microarray images (see Fig. 2.14), where the spots in the DNA sequences are detected by matching a boundary template (modelled by a circular filter) to the edge map of the image (estimated via Laplacian of Gaussian method). An alternative correlation matching algorithm is introduced by Broeke *et al.* to the vesicle tracking system in [18] for cellular traffic in a living cell (see Fig. 2.15). Each vesicle in the time-lapse images is tracked by matching the associated appearance models derived from the local intensity histogram of the vesicle in the first frame.

2.1.7 Geometrical Shape Models

In recent years, the prior shape information of target objects has been also taken into consideration for image analysis in microscopy when the geometric properties of the target objects are governed by certain observable characteristics. Some publications have proposed different

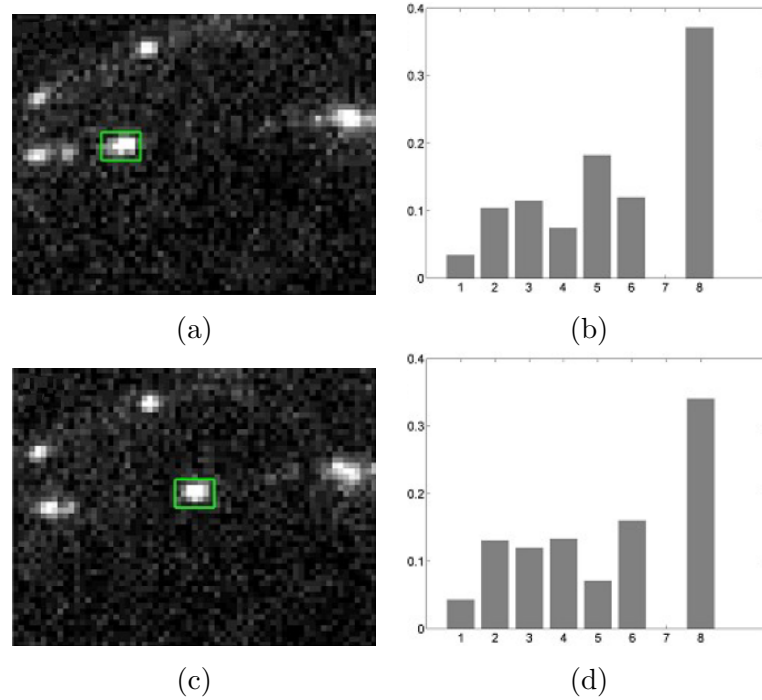


Figure 2.15: An example of the appearance model used for tracking (a) a highlighted vesicle at frame 1; (b) the appearance model of the highlighted vesicle in (a); (c) the same highlighted vesicle at frame 2; (d) the appearance model of the highlighted vesicle in (c) (Reproduced from [18])

geometrical shape models to simulate the prior shape information in mathematical terms for fitting the shapes of target objects in various applications. These models can tolerate certain variance in the object shapes, and are capable of regenerating an appropriate approximation of the modelled objects. For instance, the dynamic Gaussian mixture model (DGMM) proposed by Xiong *et al.* in [92] is a parametric shape model that fits the shape of each cell with a Gaussian distribution.

Alternatively, a doughnut shape model is proposed by Sarder *et al.* in [85] for modelling and detecting the hollow spots in a noisy microarray image. As shown in Fig. 2.16, the doughnut shape model is built by a parametric circular outer periphery with an elliptical centre hole. The model parameters are estimated from the information of the target spot and are optimized by using Gibbs sampling method.

In [3], a new cell splitting algorithm is proposed by Bai *et al.* based on concave points and ellipse fitting. The algorithm includes two parts: contour pre-processing and ellipse processing. The purpose of the contour pre-processing is to smooth the fluctuations and find the concave points of the contour of the cell clumps, and divide the contour into different fragments that may belong to different cells via the concave points. Since the single cells have similar ellipse-like shapes, the idea of the second process (the ellipse processing) is to associate an ellipse to each touching cell and the boundary of the cell can be fitted by this ellipse. Hence, the contour

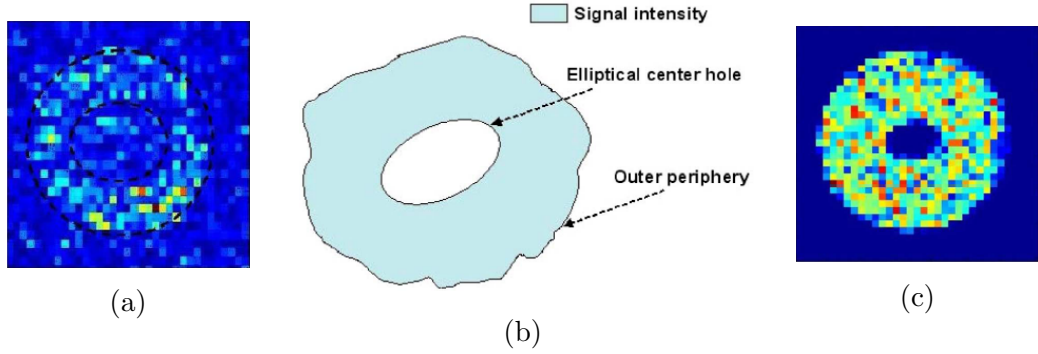


Figure 2.16: Doughnut shape modelling for gene signal estimation. (a) *Intensity image of a single hollow spot*; (b) *a schematic view of a oligonucleotide-based microarray spot*; (c) *a simulated spot-image with purified information after the shape modelling with a doughnut shape that has circular outer periphery with an elliptical centre hole* (Reproduced from [85])

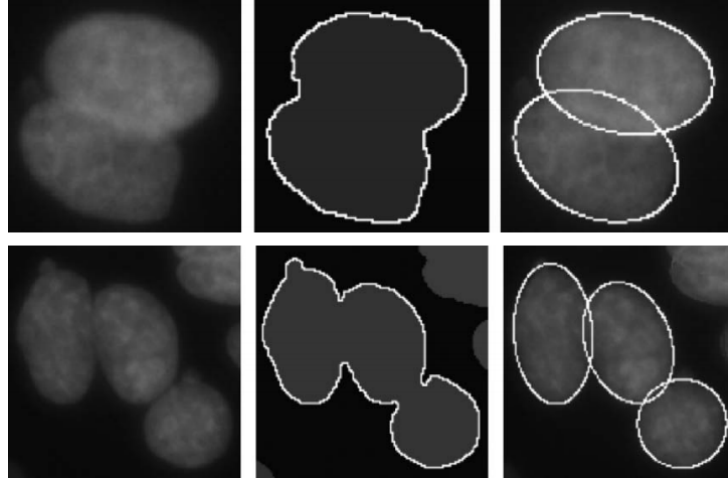


Figure 2.17: Results of the ellipse-fit cell splitting algorithm in [3]. (Reproduced from [3])

fragments that belong to the same cell are grouped by matching the properties of the ellipses fitted to each fragment. The splitting results are shown in Fig. 2.17.

A hyper-quadric model-based segmentation for isolating the nuclei in a clump structure is proposed by Cong and Parvinin [28]. The big clumps consisting of several nuclei squeezed together are firstly separated by thresholding. Then each clump is partitioned into segments that are partial nucleus boundaries. The local features of the nuclei such as the step-edge boundary segments, the roof-edge boundary segments and the concave corners are extracted from a polygonal approximation of the initial boundary segments. A step-edge is the boundary segment that belongs to one and only one nucleus while a roof-edge boundary segment is the common boundary between every two squeezed nuclei in one clump. The shape of each nucleus is then estimated by fitting a hyper-quadric model to the local features that are grouped and

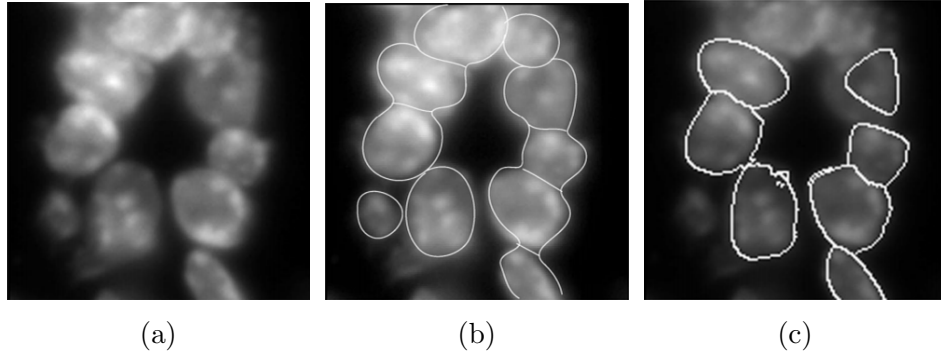


Figure 2.18: Results of the hyper-quadric model-based nuclei segmentation in [28] (a) *Original image clump of nuclei.* (b) *Ground-truth segmentation.* (c) *Results of the segmentation model.* (Reproduced from [28])

belong to the same nucleus. The grouping process is using a neighbourhood function defined over the boundary segments of the clump. The search space of the neighbourhood is expressed as an assignment matrix with an appropriate cost function to ensure local and neighbourhood consistency. Each possible configuration of a nucleus defines a path (the boundary) and the path with least overall error is selected for final segmentation. The segmentation results are shown in Fig. 2.18.

In [52], ellipsoidal models are proposed by Jung and Kim to parameterize the contours of the nuclei in the watershed-based segmentation of clustered nuclei in cervical and breast cell images. The image of clustered nuclei is firstly transformed to an inverse distance map. The markers for the watershed algorithm are then extracted as the minima in the inverse distance map by using the H -minima transform, where the all minima whose depth is lower than or equal to a given h -value are suppressed. In order to estimate the optimal h -value, a size-invariant segmentation distortion evaluation function is applied to the segmentation algorithm. The idea is to associate an ellipsoidal model to each clustered nucleus detected by particular h -value, and the ellipsoidal model is an implicit second-order polynomial computed from the partial original boundaries belong to the nucleus. The segmentation distortion evaluation function is a measure of the fitting residual between the resultant ellipse and the actual boundaries, and the optimal h -value is estimated by minimizing the function. In addition, the jagged contour problem by the watershed can be alleviated since the nuclei are described by the parameterized ellipsoidal models. The segmentation results are shown in Fig. 2.19.

2.1.8 Discussion

Thresholding based image segmentation aims to partition an input image into pixels of two or more values through comparison of pixel values with the predefined threshold value. The disadvantage of thresholding is the requirement of suitable threshold value(s), and the inconsistency of shape information that can lead to merging of nearby objects..

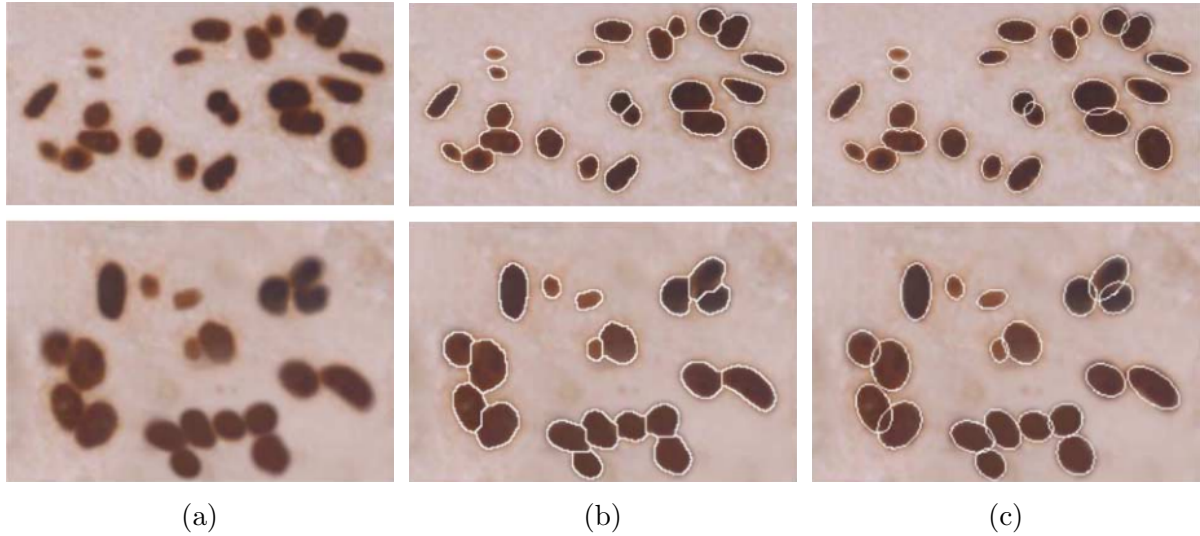


Figure 2.19: Segmentation results of the ellipsoidal model-based watershed algorithm in [52] (a) *Original cervical and breast cell images with clustered nuclei.* (b) *Segmentation results of the watershed-method.* (c) *Contour adjustment by the parameterized ellipsoidal models.* (Reproduced from [52])

Watershed segmentation can isolate the objects that are clustered in the image but it leads to over-segmentation which usually needs post-processing to solve the problem. Also, the segmented regions can lack smoothness due to the image noise and the discrete nature of image.

Clustering based segmentation attempts to identify similar groups of pixels based on selected characteristics, and provides some statistics that can yield information about the ability of each variable to differentiate the segments. However, the quality of the final solution depends largely on the initial set of clusters, and especially the manual selection of the number of clusters. An inappropriate choice of that number may yield poor results, and automated estimation can be difficult when the clusters are less distinguishable from each other.

The classification models (especially the Bayesian inference models) have attracted huge interest in image segmentation. This is because of the ability to integrate spectral, textural, contextual, spatial properties of image and even prior knowledge in the form of the prior distribution. Also, it seems to be a promising approach for unsupervised image segmentation. However, the mathematical formulation and high computational complexity are the drawbacks.

The AC model can be used to efficiently address the problem of curve (or surface) propagation in an implicit manner, and provides a direct way to estimate the geometric properties of the evolving structure. Snakes can be adopted to different requirements or conditions in various application by using different energy terms in the energy function. The drawback of the AC model is that the accuracy is governed by the convergence criteria used in the energy minimization technique. Also, snakes are usually sensitive to the initialization and noise, and often get stuck in local minima states. ASMs and AAMs are good approaches to tackle the initialization

and noise problem, however, the objects they are fitting require similar shapes to the training samples.

Correlation matching is a simple and effective shape based method for detecting the objects with uniform shape appearance, but it fails for objects that have variant shapes or change shapes in noisy environment.

Geometrical shape fitting tends to use the geometrical properties as a priori knowledge. The shape models can tolerate certain variance in the object shapes, and are capable to regenerate an appropriate approximation of the modelled objects. Also, the shape models are potentially invariant to the luminance, orientation and scale of the objects. The drawback of this approach is that there is no particular formulation and it is fully customized for the specific objects in different applications.

As described in the previous sections, image driven approaches are generally simple and can be applied to a wide class of problems. However, they provide little specific information for the segmented objects in situations where the shape characteristics of the objects of interests are well understood. On the other hand, model driven approaches can be used to obtain the shape parameters of the objects of interest which can be useful for further shape analysis. But the downsides are that the model driven approaches typically generalize poorly and are more difficult to optimize.

2.2 Problems addressed in this Work

There are three tasks in this work. The first one is to extract spot objects in the given pair of registered protein molecule images (Fig. 2.20) and measure the colocalization of the spots across the two images. The second is to segment 3D synaptic objects from a given image stack where each slice of the stack is an observation of the object at a particular depth in the brain tissue (Fig. 2.21). The last task is to segment the overlapped muscle fibre/cell objects in the skeletal muscle images (Fig. 2.22).

The images for these tasks are observed from confocal microscopes, where the target objects are stained with different fluorescent markers before the scanning. An overview of how these images are generated using confocal fluorescent microscopy is firstly given in the following section. It is then followed by a discussion of how the proposed solution in this work is related to the existing techniques.

2.2.1 Confocal Fluorescent Microscopy

Confocal fluorescent microscopy combines the fields of fluorescence microscopy and confocal microscopy. Fluorescence microscopy refers to the fluorescent staining and imaging process of the target objects but is not a specific type of microscope. For fluorescence imaging, the confocal microscopy currently prevails as the technique of choice for studying specimens which exhibit

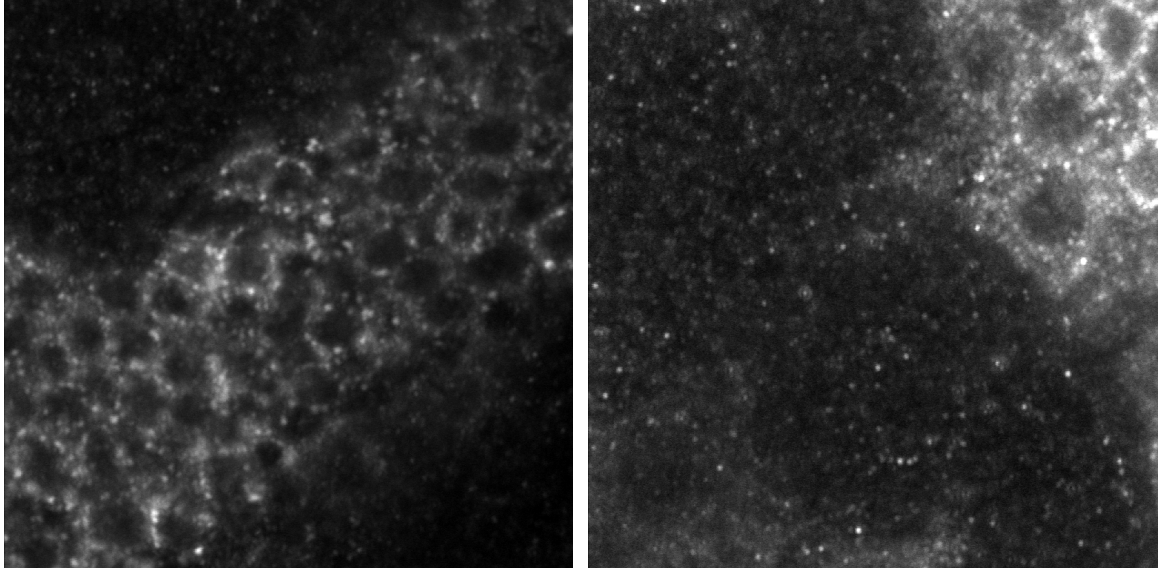


Figure 2.20: Typical registered fluorescent microscopical images with different types of spot-like proteins in each image. *(The left is a gray-scale image of stained Me31B protein particles. The right a gray-scale image of CaMKII (another type of protein). These two images are registered from the same specimen but the two different types of proteins are stained with different markers in the scanning.)*

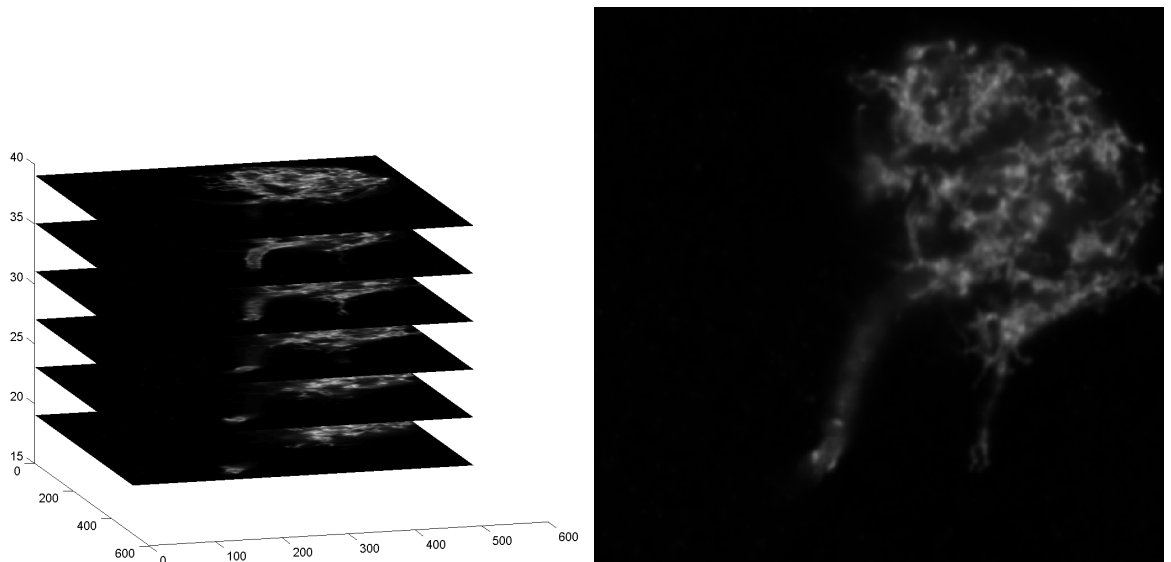


Figure 2.21: Typical image stack of 3D synapses in brain tissue. *The right shows that a slice of the stack is an observation of the synapses at a particular depth in the brain tissue.*

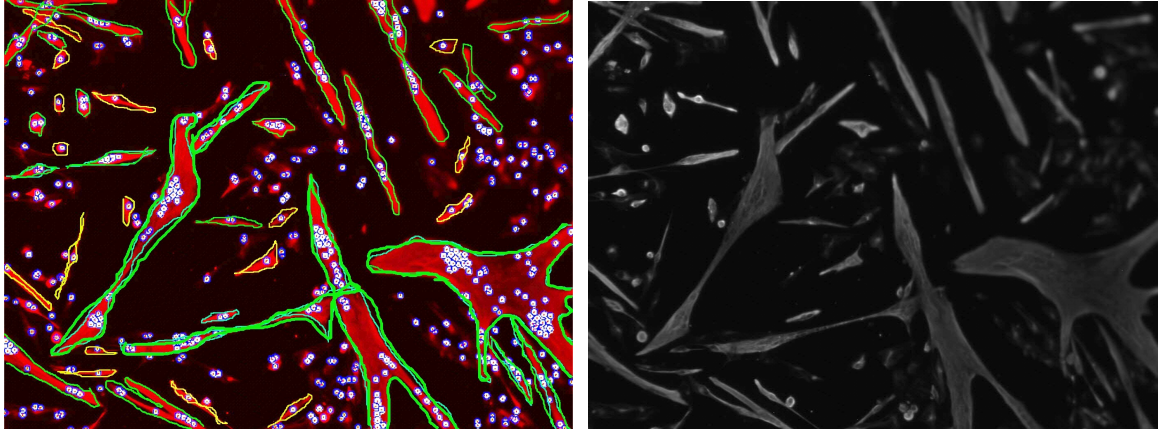


Figure 2.22: Typical skeletal muscle image with overlapped muscle fibres/cells. *The left image has a manually segmented fibres/cells. The right is a gray-scale cell-body image of the muscle fibres/cells.*

fluorescence, which allows for rapid and non-invasive scanning of three-dimensional structures.

Fluorescence Microscopy: A fluorescence microscope is an optical microscope that uses fluorescence reflection and absorption to study properties of organic or inorganic substances. In fluorescence microscopy, a sample is illuminated with light of a wavelength which excites fluorescence in the sample. The fluoresced light, which is usually at a longer wavelength than the illumination, is then imaged through a microscope objective. Two filters are normally used in this technique; an illumination (or excitation) filter which ensures the illumination is near monochromatic and at the correct wavelength, and a second emission (or barrier) filter which ensures none of the excitation light source reaches the detector.

The specimen is firstly stained with fluorophores and is then illuminated with light of a specific wavelength (or wavelengths) which is absorbed by the fluorophores, causing them to emit light of longer wavelengths (i.e. a different colour than the absorbed light). The illumination light is separated from the much weaker emitted fluorescence through the use of a spectral emission filter. Typical components of a fluorescence microscope are a light source (xenon arc lamp or mercury-vapor lamp), the excitation filter, the dichroic mirror (or dichroic beamsplitter), and the emission filter (see Fig. 2.23). The filters and the dichroic mirror are chosen to match the spectral excitation and emission characteristics of the fluorophore used to label the specimen. In this manner, the distribution of a single fluorophore (colour) is imaged at a time. Multi-colour images of several types of fluorophores must be composed by combining several single-colour images.

Confocal Microscopy: Confocal imaging was patented by Marvin Minsky [66] in 1957. The principle is to use point illumination and a pinhole in an optically conjugate plane in front of the detector to eliminate out-of-focus signal, and hence only the light produced by fluorescence

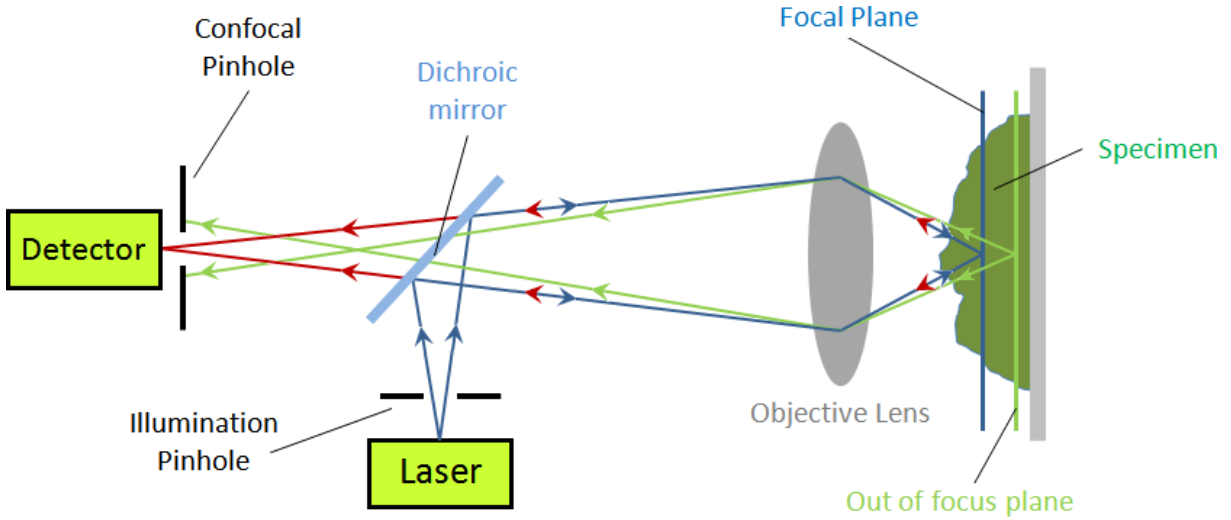


Figure 2.23: Illustration of the fluorescent confocal microscopy. *(The light emitted from the laser is firstly absorbed by the fluorescent markers stained in the specimen. The light with different wavelength is then re-emitted from the specimen and is received by the detector for the imaging process.)*

that is very close to the focal plane can be detected.

In confocal scanning microscopy, the sample is scanned by laser light and the emitted fluorescence after excitation is collected onto a photo-detector. As shown in Fig. 2.23, an aperture with a particular pinhole size is positioned in the image plane in front of the detector at a position confocal with the in-focus plane. Light emanating from the in-focus plane passes through the aperture to the detector, while artefactual emission from any out-of-focus planes (the regions above or below the focal plane) is away from the detection pinhole and hence is excluded from reaching the detector. It is this ability of the confocal microscopy that reduces the out-of-focus blur, and this confocal effect is controlled by the size of the pinhole.

To obtain a full image, the point of light is moved across the specimen by scanning mirrors. The emitted/reflected light passing through the detector pinhole is transformed into electrical signals by a photomultiplier and displayed on a computer monitor.

2.2.2 Scope for New Algorithms

Spot analysis: The first task is to determine the number and positions of the protein particles in a stained confocal microscopy image as shown in Fig. 2.20. Then it is required to measure the distribution of distances between particular proteins i.e. colocalisation analysis. The difficulty is that the particles to be analyzed do not manifest as easily isolated image patches but often appear as large and irregularly shaped particle clusters. The watershed algorithm seems a potential solution for isolating the particles. However segmentation alone does not yield particle position necessary for this analysis. As previously discussed, shape characteristics have been

investigated with the use of an elliptical bounding curve [52] and dynamic Gaussian mixture models [92]. The work presented in this thesis continues along similar lines by using 2D GMMs to fit the intensity profile of segmented spot regions in the image. The key difference to previous work however, is that the GMM is used as a functional fit to that profile and not to some features extracted from the profile as in [52]. The resulting parameterization allows each Gaussian in the mixture to capture the size and position of each constituent particle in the spot. Using GMMs for fitting in this way allows measurement despite partial occlusion of particles in each cluster. The size of each particle is related to the variance of each Gaussian and the position is taken as the location of the mode of the Gaussian. Co-localisation analysis can then proceed using these measurements. The details of this measurement process is described in *Chapter 3* and *Chapter 4*.

3-D shape reconstruction: The second task is to segment the in-focus synaptic structures in stained confocal microscopy image stacks (as shown in Fig. 2.21) when a degree of out-of-focus detail is still present in each image of the stack. Due to the out-of-focus problem, the boundaries of the synaptic structures are difficult to define in the images. The goal of this task is to distinguish the in-focus synaptic features from the out-of-focus features. However, the segmentation becomes complicated when the luminance of the synaptic object is irregular. This is due to the defect of the fluorescent staining, where the markers are distributed unevenly over the synapses. To solve this problem, the brightness information needs to be combined with the sharpness information which is invariant to the luminance. Neither the AC model or the geometrical model is appropriate as the shape of synapses is too complex. The watershed algorithm may be a solution, but the major concern is over-segmentation which requires a second step for region merging, or makes use of specific markers [24,51,90] which have been either manually defined by the user or determined automatically with morphological operators. Moreover, the watershed algorithm does not give a nice shape definition for the segmented objects. Instead, since the decision of detecting the in-focus features is based on the sharpness of the features, the luminance of each feature across the slices, and the connectivity properties of the synaptic structure, a Bayesian inference model seems suitable to integrate the different information and segment the object. Furthermore, the method for extracting the sharpness information is informed by the work of Izeddin *et al.* [49], where the molecules are detected by the contrast at particular wavelet scale. Details are presented in *Chapter 5*.

Segmentation of overlapped objects in muscle fibre images: The third task is to segment the objects in a given image (as shown in Fig. 2.22) and classify each segmented object as a fibre or a cell based on the number of nuclei inside the object. Since the detection of candidate nuclei in each nucleus cluster is similar to the protein particles detection in the first task, the number of the candidate nuclei can be determined by using the GMM fitting algorithm proposed in the first part. However, the fibres lie in bundles in which each overlap with the other, and each has quite varied shape. Hence a single isolated shape might contain one or more fibres.

The problem is to design a method capable of resolving the fibres in the bundle at the same time as counting the nuclei associated with each fibre. Fibres tend to overlap only in a section along their length. So the problem can be refined into that of extending a detected fibre region across these "junctions" onto the continuation of the same fibre on the other side of the junction. Because the goal is to count the number of cells and fibres in each overlap region which differ by the number of nuclei, the interest of the segmentation is to find the approximate shape of each cell/fibre candidate that 'covers' the right nuclei, rather than getting the exact shapes of the objects.

Since there is no information available in the overlap area of the regions for the shape reconstruction of the occluded objects, the image driven approaches [1, 7, 24, 26, 35, 51, 84] and the classification models [11, 57, 71] are inappropriate for this task. Also, the algorithms that are based on reference samples [18, 19, 29, 30, 38] are not suitable when the shapes of fibres are highly variable. Instead, the shape of the occluded regions can only be inferred from the shapes of the non-overlapped parts of the occluded objects. Although geometric shape fitting has this ability, the variation in the shape of the target objects is too large for the parametrization. A reasonable approach is to fit an AC model to each non-overlapped part when it can efficiently address the problem of curve propagation. Since the snake is a deformable curve, the shape of occluded objects can be estimated by the region covered by the associated snake. The details are presented in *Chapter 6*.

3

GMM Shape Modelling for Spot Detection ¹

The formation of various memories require the presence of different specific messenger ribonucleotide protein (mRNP) in the neuron cells, where a mRNP is formed by the messenger ribonucleic acid (mRNA) bound by some protein particles. The development of fluorescent proteins and high resolution fluorescence imaging allow the biologists to locate the different mRNPs in a living specimen by the co-localization of the differently labelled specific protein markers. However, the quantitative interpretation of the labelled proteins is still heavily reliant on manual evaluation. In order to facilitate these studies, a new algorithmic solution is introduced in this chapter for spot detection and co-localization analysis in microscopy. The presented method is robust to substantial uncertainties in spot finding caused by variation in the positions and sizes of spots and by the presence of artefacts and background noise in the images observed from confocal microscopes. The method is capable of processing large numbers of spots automatically and effectively. Before getting into the detail of the spot detection method, a brief biological background of this work is given.

3.1 Introduction to the Biology

As shown in Fig. 3.1, a neuron cell has a main body and a dendrite that branches out from the main body. The main body contains the nucleus of the cell and the dendrite consists of the synapses (the tips). The neuron cells stimulate each other through the connected synapses in order to induce different brain activities. The formation of long-term memory (LTM) and the

¹This work is presented in part in [46, 77, 78]

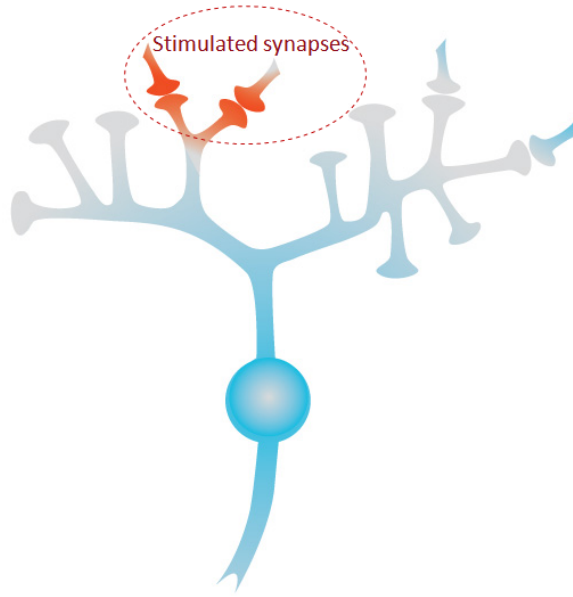


Figure 3.1: Neuron Cell *Memory formation is modulated by protein synthesis, i.e. mRNA translation, at the stimulated synapses (red) of neuron cell.*

underlying persistent synaptic changes, at cellular level, are designated by protein synthesis at stimulated synapses (in red) of the neuron cells (Fig. 3.1). The protein synthesis is ‘regulated’ by binding some repressing proteins (the repressors) to the messenger ribonucleic acid (mRNA) that is emitted from the nucleus. The repressors effectively act as switches for protein synthesis. The ideas of protein synthesis and the synthesis regulation are given in *Appendix A*.

The repressing proteins are important in the formation of memory. However, the composition and characteristics of the different proteins are still poorly understood in mature nervous systems. The research topic explored in this collaboration is the investigation of the roles of different repressing proteins in memory formation and focuses on the Me31B protein. This protein is found to be essential for different stages of the translational regulation of mRNAs in *Drosophila* (fruit flies) [6]. The biological study of the Me31B protein considers not only the distribution of the protein particles in the RNA granules (clusters of mRNAs, repressors and other proteins) located at synapses, but also the connection with other proteins involved in memory formation. The connection is derived from observing co-localization of the Me31B protein particles with particles of other types in the RNA granules.

Co-localization is measured by analysing pictures taken from the synapses of the neuron cell in brain tissue of *Drosophila*. Fig. 3.2 shows an example of the pictures. In order to visualize the Me31B from the dendritically localized RNA granules, the Me31B protein and one of the other proteins are double-stained by different fluorescent markers. The brain tissue is fixed and scanned by the confocal microscope (model *Zeiss LSM 510 meta*). Since the light emitted from the different fluorescent markers has different wavelength, the gray-scale images

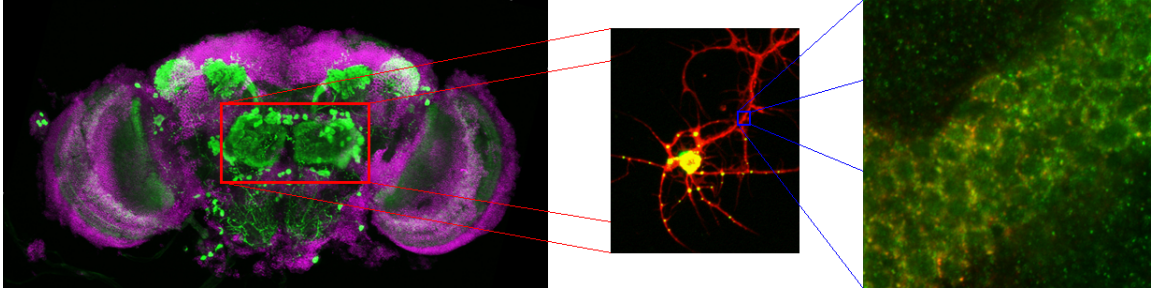


Figure 3.2: Drosophila neuronal RNA-Granules (*The left is an image of a Drosophila head, the middle shows a neuron cell from the Drosophila brain, and the right shows an image of the double-stained particles in RNA granules at a synapse of the neuron cell.*)

(Fig. 3.3(b)&(c)) for the different types of proteins are registered at the same time by detecting the light with the corresponding wavelengths. This means the two images have no offset, and a pixel $\vec{x}$ in one image is projected from the same actual location as the pixel $\vec{x}$ in the other image. Hence, co-localization analysis of the different proteins can rely on the positions of the protein particles in the images. The image shown in Fig. 3.3(a) is a superimposed image formed by assigning the gray-scale images of Me31B and CaMKII (one of other types) to the red and green channels respectively.

3.2 Introduction to Analysis

The analysis of the distribution and co-localization of the specific proteins requires the detection of the protein particles in the observed images and the parameterized geometric characteristics (locations and shapes) of the detected particles. As shown in Fig. 3.3, the images contain a huge amount of spots of varying shapes associated with the protein particles. Some small spots with regular shapes correspond to individual protein particles. Large and irregular spots are formed by the merged appearance of clusters of particles (cluster spots). The manual detection and measurement of the particles are impractical for such innumerable irregular spot shapes. Instead, an image processing tool that can automatically analyse large amount of particles is preferable. However, the detection and parametrization are complicated by the presence of cluster spots. The difficulty of analysing the particles inside the clusters is that the amount and sizes of the cluster candidates can not be easily determined from the ‘merged’ appearance in the image. Moreover, the actual particles are arranged in a 3D structure, and due to the physical limit of the confocal microscope, the objects projected into the 2D image may not only be the actual object at the imaging ‘depth’ in the specimen, but also the objects located at lower depths. The objects at lower depths can appear smaller than the usual minimum particle size and hence the clusters may contain a greater number of particles than expected simply from their appearance.

The core problem is to estimate the number and shapes of the particles present in the images.

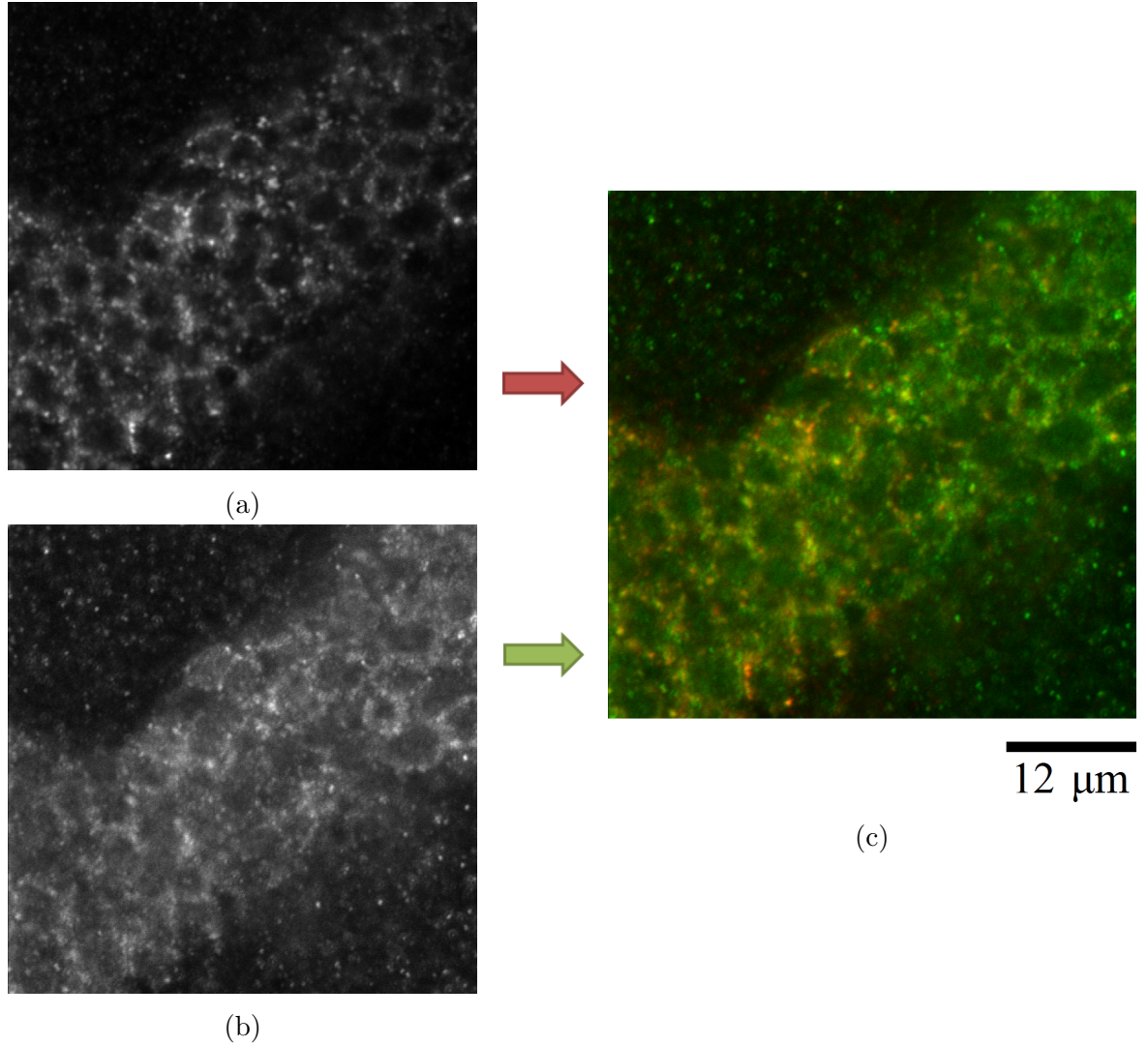


Figure 3.3: Images of double-stained particles of two different types of proteins. (a) a gray-scale image of stained Me31B particles that assigned to the red channel of (c); (b) a gray-scale image of CaMKII (one of other types of protein) in the RNA granules that assigned to the green channel of (c); (c) an RGB image of double-stained particles which is generated assigning the gray-scale images (a&b) of two different proteins respectively to the red and green channels. The images are scanned by a Zeiss LSM 510 confocal microscope.

As shown in Fig. 3.4, the intensity profile of a spot associated with a single protein particle in the given image has a bell shape. Also, since the shapes of the spots are affected by the point-spread-function of the confocal microscope during the scanning, and the point-spread-function can be approximated by Gaussian distribution [97], an assumption is then made that each protein particle in the image can be modelled by a 2D Gaussian distribution. Hence clusters can be modelled with a Gaussian mixture model so that the mixture components represent the corresponding particles in the cluster. Therefore, the goal of this work is to fit a Gaussian

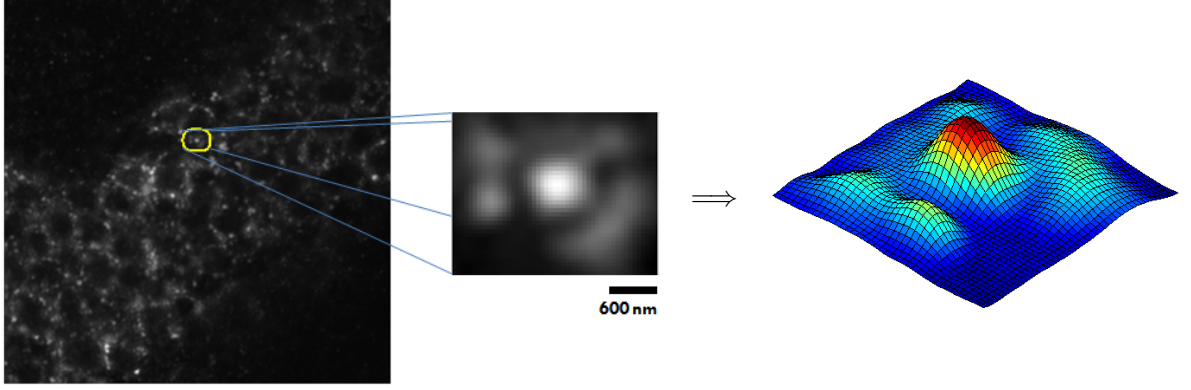


Figure 3.4: Brightness profile of image sample. (The left image is an observed image of the Me31B protein in a test sample shown in Fig. 3.3(b). A zoom-in of the image consists of a cluster of the protein particles is given in the middle, and the corresponding brightness profile is shown on the right.)

mixture model (GMM) to the intensity profile $\mathbf{I}$ of a given particle cluster as follows:

$$I(\vec{x}) = \sum_{m=1}^M \alpha_m N(\vec{x} | \vec{\mu}_m, \Sigma_m) \quad (3.1)$$

where $I(\vec{x})$ is the intensity value at pixel site $\vec{x} = (h, k)$ in the image. α_m is the mixing weight of the m^{th} component of GMM which has 2D Gaussian distribution $N(\vec{x} | \vec{\mu}_m, \Sigma_m)$ with mean $\vec{\mu}_m$ and covariance Σ_m . Also, the mixing weight α is normalized so that $\sum_{m=1}^M \alpha_m = 1$.

Accordingly, each spot (both the single particles and clusters) in the given image is parameterized by a GMM with parameters $\Theta = \{M, \alpha_1, \dots, \alpha_M, \theta_1, \dots, \theta_M\}$, where M is the number of the components in the GMM which represents the amount of the cluster candidates, and $\theta_m = \{\vec{\mu}_m, \Sigma_m\}$ the parameter set of the m^{th} component where $\vec{\mu}_m$ and Σ_m record the location and shape characteristics of the particle.

The GMM shape modelling approach is essentially a function fitting approach. As the spot detection and parametrization problems become a shape modelling problem, the locations and the sizes of the particles of both channels are then analysed by estimating the parameters Θ of the corresponding GMMs. This enables the analysis of colocalization of particles across the red and green channels of a given double-stained image.

In this case, the initialization of each GMM is based on the peaks of the corresponding spot (after smoothening) as the mode of a 2D Gaussian is where the peak of the distribution is located, and the model parameters (means, covariances etc.) are initialized using heuristics based on the intensity profile of the spot regions. However, the number of particles can be larger than the number of peaks in the spot of clustered particles. This makes the optimization process difficult as both the number M of components and the distribution parameters are needed to be estimated simultaneously. Even though joint optimisation of the number of components and shape parameters for clearly isolated spot regions may seem like overkill, in practice quick convergence in this scenario means that applying a simpler model would not offset the overhead

of a pre-process to detect the isolated spots. Although fitting the intensity surface is a novel application of the GMM, the idea of GMM modelling has been widely adopted for data point clustering. Hence, the existing techniques for solving the optimization problem in the data clustering can be used as references for solving the problem in this work. The technique adopted in this work is the Split-and-Merge Expectation-Maximization (SMEM) algorithm. Several alternative SMEM techniques [60, 63, 73, 99] have been published in the state-of-art of signal processing. However, since these techniques are designed for data point clustering, a new SMEM algorithm that is based on the ‘target’ shape is more suitable for this work. Therefore, a novel technique is developed in this work that uses the estimation error between the GMM and the shape of the cluster and is named the error-based SMEM (eSMEM) algorithm.

In this chapter, before getting into the details of the eSMEM algorithm, the idea of conventional expectation maximization (EM) algorithm for GMMs is firstly shown in *Section 3.3*. A discussion of the SMEM algorithm and a review of the alternative intrinsic split methods are given in *Section 3.4*, and finally the new eSMEM algorithm is described in *Section 3.5*.

3.3 EM Algorithm for GMM Parameter Estimation

The problem is to estimate the GMM parameters of each spot shape, given only the brightness profile. One of the possible solutions is the expectation-maximization (EM) algorithm [10, 12]. The EM algorithm is a Maximum Likelihood (ML) method that iteratively updates the model parameter estimates in the presence of hidden or missing variables.

Each iteration of the EM algorithm consists of two steps, the expectation step (E-step) and the maximization step (M-step). The E-step is the first step in which the hidden data are estimated given the observed data and current estimate of the model parameters. In the M-step, the likelihood function is maximized under the assumption that the hidden data are known and set by the estimates from the E-step. The parameter estimates are guaranteed to converge since the algorithm increases the likelihood of the estimates at each iteration.

3.3.1 Likelihood

Given a set of N observed data $\vec{X} = \{\vec{x}_1, \dots, \vec{x}_N\}$, the parameter estimate Θ of a mixture density model is based on the probability density function $p(\vec{X}|\Theta)$ as follows

$$p(\vec{X}|\Theta) = \prod_{n=1}^N p(\vec{x}_n|\Theta) \quad (3.2)$$

where $p(\vec{x}_n|\Theta)$ is the probability of the data $\vec{x}_n$ realized from the mixture model as follows

$$p(\vec{x}_n|\Theta) = \sum_{m=1}^M \alpha_m p_m(\vec{x}_n|\theta_m) \quad (3.3)$$

where $p_m(\vec{x}_n|\theta_m)$ is the probability of the data $\vec{x}_n$ realized from the m^{th} mixture component with kernel parameters θ_m .

The EM algorithm interactively updates the estimate for Θ so that the log likelihood function $\mathcal{L}(\Theta) = \ln p(\vec{X}|\Theta)$ is maximized. The goal of each iteration is to update the estimate of Θ so that the likelihood $\mathcal{L}(\cdot)$ is increased from the current estimate $\Theta^{(i)}$. This is equivalent to maximizing the difference of the likelihoods,

$$\mathcal{L}(\Theta) - \mathcal{L}(\Theta^{(i)}) = \ln p(\vec{X}|\Theta) - \ln p(\vec{X}|\Theta^{(i)}) \quad (3.4)$$

3.3.2 Hidden Variable

In order to make the ML estimate tractable, a set of hidden variables $Y = \{y_1, \dots, y_M\}$ is introduced where y_m is the value representing the component of the mixture that ‘generates’ the data point. Therefore, the total probability $p(\vec{X}|\Theta)$ of the mixture model can be expressed in terms of Y as

$$p(\vec{X}|\Theta) = p(\vec{X}|Y, \Theta)p(Y|\Theta) = \sum_{m=1}^M p(\vec{X}|y_m, \Theta)p(y_m|\Theta) \quad (3.5)$$

According to [12], Eq. (3.4) can be rewritten as

$$\mathcal{L}(\Theta) - \mathcal{L}(\Theta^{(i)}) = \ln \left(\sum_{m=1}^M p(\vec{X}|y_m, \Theta)p(y_m|\Theta) \right) - \ln p(\vec{X}|\Theta^{(i)}) \quad (3.6)$$

This equation can be simplified using Jensen’s inequality [50] and becomes

$$\mathcal{L}(\Theta) - \mathcal{L}(\Theta^{(i)}) \geq \sum_{m=1}^M p(y_m|\vec{X}, \Theta^{(i)}) \ln \left(\frac{p(\vec{X}|y_m, \Theta)p(y_m|\Theta)}{p(y_m|\vec{X}, \Theta^{(i)})p(\vec{X}|\Theta^{(i)})} \right) \quad (3.7)$$

The derivation of Eq. (3.7) from Eq. (3.6) can be found in [12]. The general form of the EM algorithm can then be expressed according to [12] as

$$\Theta^{(i+1)} = \arg \max_{\Theta} \left\{ \mathcal{L}(\Theta^{(i)}) + \sum_{m=1}^M p(y_m|\vec{X}, \Theta^{(i)}) \ln \left(\frac{p(\vec{X}|y_m, \Theta)p(y_m|\Theta)}{p(y_m|\vec{X}, \Theta^{(i)})p(\vec{X}|\Theta^{(i)})} \right) \right\} \quad (3.8)$$

Since this expression is only with respect to Θ , the terms that do not consist of Θ can be considered as fixed values and hence removed from optimization. Therefore the general form of the EM algorithm in Eq. (3.8) can be simplified as

$$\begin{aligned} \Theta^{(i+1)} &= \arg \max_{\Theta} \left\{ \sum_{m=1}^M p(y_m|\vec{X}, \Theta^{(i)}) \ln p(\vec{X}, y_m|\Theta) \right\} \\ &= \arg \max_{\Theta} \left\{ \mathcal{E}_{Y|\vec{X}, \Theta} \{ \ln p(\vec{X}, y_m|\Theta) \} \right\} \end{aligned} \quad (3.9)$$

where $\Theta^{(i+1)}$ is the updating values for Θ . The goal of the E-step at each iteration is to determine the conditional expected value of the log likelihood, $\mathcal{E}_{Y|\vec{X}, \Theta} \{ \ln p(\vec{X}, y_m|\Theta) \}$, and the M-step maximizes the expectation with respect to Θ .

3.3.3 Parameter Estimation

In order to compute the parameter estimates $\Theta^{(i+1)}$ for the GMM so that the conditional expectation of the EM algorithm is optimized, the expression in Eq. (3.9) is simplified according to [10] as follows

$$\begin{aligned}\Theta^{(i+1)} &= \arg \max_{\Theta} \left\{ \mathcal{E}_{Y|\vec{x}, \Theta} \{ \ln p(\vec{X}|y_m, \Theta) \} \right\} \\ &= \arg \max_{\Theta} \left\{ \sum_{m=1}^M \sum_{n=1}^N \ln(\alpha_{y_m}) p(y_m|\vec{x}_n, \Theta^{(i)}) + \sum_{m=1}^M \sum_{n=1}^N \ln [p_{y_m}(\vec{x}_n|\theta_{y_m})] p(y_m|\vec{x}_n, \Theta^{(i)}) \right\}\end{aligned}\quad (3.10)$$

Since y_m represents the labels of the mixture components, α_{y_m} is the mixing weight α of the component y_m , and $p_{y_m}(\vec{x}_n|\theta_{y_m})$ is similar to $p_m(\vec{x}_n|\theta_m)$ in Eq. (3.3) that represents the probability of the data $\vec{x}_n$ realized from the component y_m . Also, $p(y_m|\vec{x}_n, \Theta^{(i)})$ is a posterior probability density for y_m from observed data $\vec{x}_n$ and given parameters $\Theta^{(i)}$, the density function is stated as

$$p(y_m|\vec{x}_n, \Theta^{(i)}) = \frac{p(\vec{x}_n|y_m, \Theta^{(i)})p(y_m|\Theta^{(i)})}{p(\vec{x}_n|\Theta^{(i)})} = \frac{p_{y_m}(\vec{x}_n|\theta_{y_m}^{(i)})\alpha_{y_m}^{(i)}}{\sum_{l=1}^M \alpha_l^{(i)} p_l(\vec{x}_n|\theta_l^{(i)})} \quad (3.11)$$

Note that $p(\vec{x}_n|y_m, \Theta^{(i)})$ is the probability of $\vec{x}_n$ being observed from the y_m^{th} component in the mixture density model with parameters estimate $\Theta^{(i)}$, and hence is equivalent to $p_{y_m}(\vec{x}_n|\theta_{y_m}^{(i)})$ which is the probability density of the y_m^{th} mixture component defined in Eq. (3.3). Also, $p(y_m|\Theta^{(i)})$ is the probability of the y_m^{th} component from the mixture density model and is simply assigned by the mixing weight α_{y_m} of the component.

The maximization of the expression in Eq. (3.10) is equivalent to maximizing the term containing α_{y_m} and the term containing θ_{y_m} independently. The expression for $\alpha_{y_m}^{(i+1)}$ is then derived from the first term by differentiation with respect to α_{y_m} . Similarly, the expressions for $\mu_{y_m}^{(i+1)}$ and $\Sigma_{y_m}^{(i+1)}$ are derived from the second term by differentiating the term that substituted with Gaussian density with respect to each of the parameter respectively. Consequently, according to [10], the updated values for the parameters in the EM iterative process are

$$\begin{aligned}p(y|\vec{x}_n, \Theta^{(i)}) &= \frac{\alpha_y^{(i)} N_l(\vec{x}_n|\theta_y^{(i)})}{\sum_{m=1}^M \alpha_m^{(i)} N_m(\vec{x}_n|\theta_m^{(i)})} \\ \alpha_y^{(i+1)} &= \frac{1}{N} \sum_{n=1}^N p(y|\vec{x}_n, \Theta^{(i)}) \\ \mu_y^{(i+1)} &= \frac{\sum_{n=1}^N \vec{x}_n p(y|\vec{x}_n, \Theta^{(i)})}{\sum_{n=1}^N p(y|\vec{x}_n, \Theta^{(i)})} \\ \Sigma_y^{(i+1)} &= \frac{\sum_{n=1}^N p(y|\vec{x}_n, \Theta^{(i)}) (\vec{x}_n - \mu_y^{(i+1)}) (\vec{x}_n - \mu_y^{(i+1)})^T}{\sum_{n=1}^N p(y|\vec{x}_n, \Theta^{(i)})}\end{aligned}\quad (3.12)$$

where $\alpha_y^{(i+1)}$, $\mu_y^{(i+1)}$ and $\Sigma_y^{(i+1)}$ are the updated parameters for the component $y \in [1, M]$. An example of the EM process for data points clustering is given in Fig. 3.5

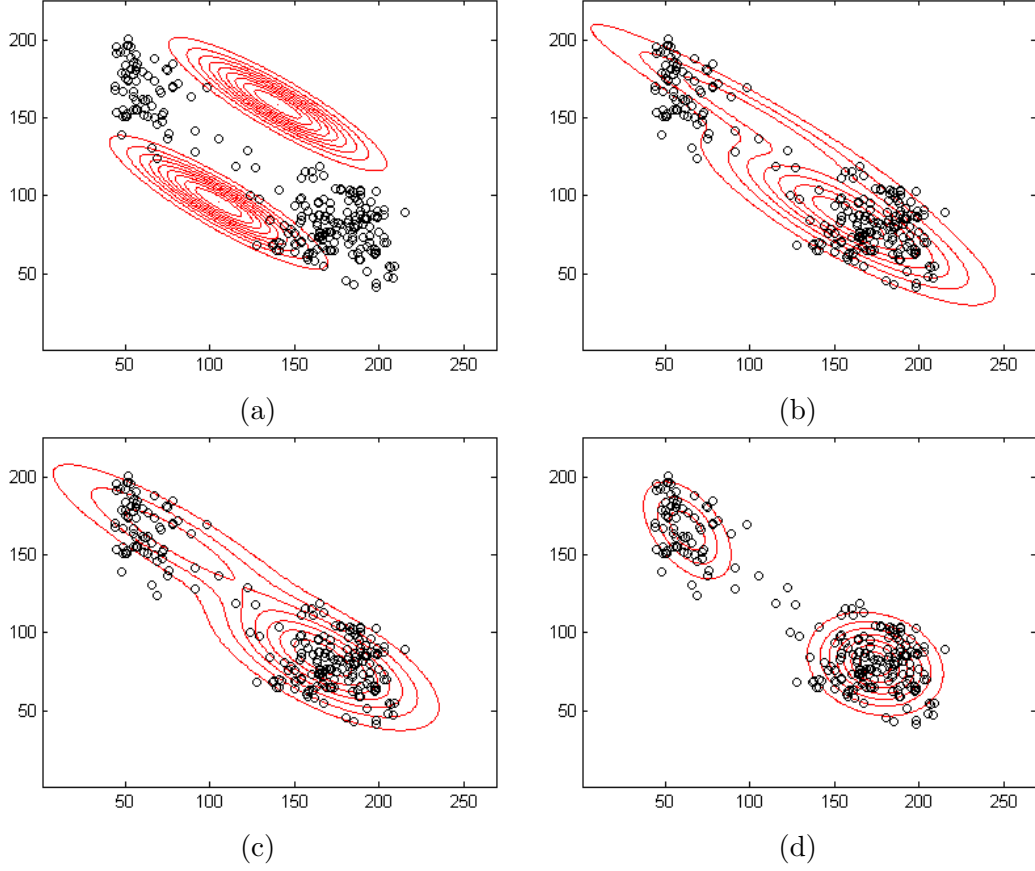


Figure 3.5: EM for sample data clustering. The EM process starts at (a) with $\alpha_1^0 = \alpha_2^0 = 0.5$, $\mu_1^0 = [96, 105]^T$, $\mu_2^0 = [159, 140]^T$ and $\Sigma_1^0 = \Sigma_2^0 = \begin{bmatrix} 400 & -540 \\ -540 & 900 \end{bmatrix}$. The process then increases the likelihood of the estimation in each iteration shown in (b) & (c). The converged estimation after 9 iterations is shown at (d) with $\alpha_1^0 = 0.69$, $\alpha_2^0 = 0.31$, $\mu_1^0 = [80, 174]^T$, $\mu_2^0 = [167, 61]^T$, $\mu_2^0 = [167, 61]^T$, $\Sigma_1^0 = \begin{bmatrix} 290.5 & -87.5 \\ -87.5 & 422.2 \end{bmatrix}$ and $\Sigma_2^0 = \begin{bmatrix} 336.2 & -97.8 \\ -97.8 & 179.3 \end{bmatrix}$.

The EM algorithm requires an initialization $\Theta^0 = \{M^0, \alpha_1^0, \dots, \alpha_M^0, \theta_1^0, \dots, \theta_M^0\}$ of the parameters Θ for starting the estimation process. The detail of parameter initialization for shape modelling is given in Section 3.5.2. Also, the iterative process requires a termination scheme. By defining $\mathcal{E}^{(i)}$ as the conditional expectation $\mathcal{E}_{Y|\vec{X}, \Theta} \{\ln p(\vec{X}|y_m, \Theta^{(i)})\}$ from Eq. (3.10) for given $\Theta^{(i)}$, the process is halted when the rate of convergence $\frac{\mathcal{E}^{(i)} - \mathcal{E}^{(i-1)}}{\mathcal{E}^{(i-1)}}$ is small enough and $\Theta^{(i)}$ is the final output.

3.3.4 Limitation of the EM Algorithm

As given in Eq. (3.12), the classic EM algorithm is not capable of estimating the number M of components, but only optimizing the distribution parameters $\{\alpha_1, \dots, \alpha_M, \theta_1, \dots, \theta_M\}$ for GMM.

In fact, the mixture model remains the same number of components that are specified by the user in initialization through the whole process, i.e. $M = M^0$. However, in this application, the number of the mixture components is not known a priori. The technique that extends the EM algorithm for estimating the model order (M) in GMM estimation is well established [73, 99], and is called the split and merge expectation maximization (SMEM) algorithm. A review of the SMEM algorithm is given in the following section.

3.4 The SMEM Algorithm

The idea of the SMEM algorithm is to introduce a split and merge operation to alter the model order (M) of the original parameter estimate Θ^* obtained from the initial EM process. This operation is iterative and consists of two simultaneous steps (the merge step and the split step) in each iteration. In the merge step, two mixture components i and j are selected from Θ^* by using a merge criterion, and are merged into a new component k' in order to form a new parameter set Θ'_{merge} with lower M . On the contrary, the split step selects a component k selected from Θ^* by using a split criterion, and split k into two new components i' and j' form a new parameter set Θ'_{split} with higher M . Both Θ'_{merge} and Θ'_{split} are optimized by the EM process and converge to Θ^*_{merge} and Θ^*_{split} respectively. The new estimates Θ^*_{merge} and Θ^*_{split} are compared with Θ^* by the corresponding conditional expectations $\mathcal{E}^*_{merge}$, $\mathcal{E}^*_{split}$ and $\mathcal{E}^*$ respectively. The estimate with higher value of $\mathcal{E}$ is subjected to the next iteration as Θ^* . Similar to the EM process, the split and merge operation is terminated when the rate of convergence of updating $\mathcal{E}^*$ is less than a certain value and the corresponding parameter estimate Θ^* is the final output the SMEM process.

3.4.1 Basic Split and Merge

The SMEM algorithm was firstly presented by Ueda *et al.* in [73]. The merge step in [73] sets the parameters $\{\alpha_{k'}, \theta_{k'}\}$ of the new component k' as linear combinations of the merge candidates i and j with parameters $\{\alpha_i, \theta_i\}$ and $\{\alpha_j, \theta_j\}$ respectively as follows

$$\begin{aligned}\alpha_{k'} &= \alpha_i + \alpha_j \\ \theta_{k'} &= \frac{\alpha_i \theta_i + \alpha_j \theta_j}{\alpha_{k'}}\end{aligned}\tag{3.13}$$

The split step in [73] estimates the parameters $\{\alpha_{i'}, \vec{m}u_{i'}, \Sigma_{i'}\}$ and $\{\alpha_{j'}, \vec{m}u_{j'}, \Sigma_{j'}\}$ respectively of the new component i' and j' from the split component k with parameters $\{\alpha_k, \vec{m}u_k, \Sigma_k\}$ by

simply reversing the merge method defined in Eq. (3.13) as follows

$$\begin{aligned}\alpha_{i'} &= \alpha_{j'} = \frac{\alpha_k}{2} \\ \vec{\mu}_{i'} &= \vec{\mu}_k + \epsilon \\ \vec{\mu}_{j'} &= \vec{\mu}_k + \epsilon' \\ \Sigma_{i'} &= \Sigma_{j'} = (\det(\Sigma_k))^{\frac{1}{2}} \mathbf{I}_2\end{aligned}\tag{3.14}$$

where ϵ and ϵ' are some small and random perturbation vector, and the last equation estimates the covariance matrices to be square root of determinant of Σ_k while $\mathbf{I}_2$ is a 2×2 identity matrix. The idea of the Ueda *et al.* SMEM algorithm is adopted in the dynamic Gaussian mixture model proposed by Xiong *et al.* in [92] for cell tracking in time-lapse image sequence.

Latterly, in [99], Zhang *et al.* argued that there is a lack of dependence in the estimates of the mean $\vec{\mu}$ and covariance Σ provided by the SMEM algorithm of Ueda *et al.*, when $\vec{\mu}$ and Σ are the 1st moment and the 2nd moment respectively of the Gaussian distribution. To relate the two parameters in the estimation, Zhang *et al.* also introduced a different merge method and two alternative split methods in [99], which are briefly reviewed in the following sections.

3.4.2 Modified Merge

The merge method presented in [99] is based on the assumption that the probability of the new merged component k' on data $\vec{x}$ is the sum of the probabilities of the two merging components i and j as follows

$$\alpha_{k'} p(\vec{x}|k') = \alpha_i p(\vec{x}|i) + \alpha_j p(\vec{x}|j)\tag{3.15}$$

where $p(\cdot)$ is the probability density of the corresponding mixture component. As a result, the merge equations for $\alpha_{k'}$ and $\mu_{k'}$ remain unchanged from [73], but $\Sigma_{k'}$ is estimated in terms of the means $\{\mu_i, \mu_j, \mu_{k'}\}$ and defined as

$$\begin{aligned}\alpha_{k'} &= \alpha_i + \alpha_j \\ \mu_{k'} &= \frac{\alpha_i \mu_i + \alpha_j \mu_j}{\alpha_{k'}} \\ \Sigma_{k'} &= \frac{1}{\alpha_{k'}} (\alpha_i (\Sigma_i + \mu_i \mu_i^T) + \alpha_j (\Sigma_j + \mu_j \mu_j^T)) - \mu_{k'} \mu_{k'}^T\end{aligned}\tag{3.16}$$

3.4.3 Modified Split

The split step is a reverse process to the merge step, and is clearly underdetermined as the unknown parameters outnumber the known parameters. The difficulty of designing the split method is that there are multiple feasible split solutions to the underdetermined problem. The split methods proposed in [99] are designed based on a typical constraint. The constraint is made that when the estimation equations for the parameters of $\{i', j'\}$ split from k are substituted to

the merge equations in Eq. (3.16) as merging $\{i', j'\}$ back to k , the merge estimate must be the same as the original parameters of k .

Generally, the mixing weight α_k of a component k is split by a random variable γ where $\gamma \in (0, 1)$. Then the new mixing weight $\alpha_{i'}$ and $\alpha_{j'}$ of the new components i' and j' are estimated by

$$\alpha_{i'} = \alpha_k \gamma, \quad \alpha_{j'} = \alpha_k (1 - \gamma) \quad (3.17)$$

For the distribution parameters, since Σ is a symmetric positive definite matrix, the estimation of the new means $\{\vec{\mu}_{i'}, \vec{\mu}_{j'}\}$ and covariances $\{\Sigma_{i'}, \Sigma_{j'}\}$ are based on the decomposition of the covariance matrix Σ_k . The two alternative Zhang *et al.* split methods differ only by the decomposition techniques, where the first one uses the singular value decomposition (SVD), and the second one uses the Cholesky decomposition.

3.4.3.1 Singular Value Decomposition

Because each covariance matrix is symmetric positive definite, the first split method is started by decomposing the covariance matrix of component k using singular value decomposition. Hence, Σ_k is expressed as $\Sigma_k = U \Lambda V^T$ where $\Lambda = \text{diag}[\lambda_1, \lambda_2]$ is a diagonal matrix with nonnegative diagonal elements λ in a descending order, and U and V are two orthogonal matrices. By setting $A = U \sqrt{\Lambda}$ and $A = [\vec{a}_1, \vec{a}_2]$ where $\vec{a}_1$ is the first column vector of A , the parameters of the new components i' and j' are estimated by

$$\begin{aligned} \vec{\mu}_{i'} &= \vec{\mu}_k - \eta \left(\sqrt{\frac{\alpha_{j'}}{\alpha_{i'}}} \vec{a}_1^{(k)} \right) \\ \vec{\mu}_{j'} &= \vec{\mu}_k - \eta \left(\sqrt{\frac{\alpha_{i'}}{\alpha_{j'}}} \vec{a}_1^{(k)} \right) \\ \Sigma_{i'} &= \frac{\alpha_{j'}}{\alpha_{i'}} \Sigma_k + \left((\beta - \beta \eta^2 - 1) \frac{\alpha_k}{\alpha_{i'}} + 1 \right) \vec{a}_1^{(k)} (\vec{a}_1^{(k)})^T \\ \Sigma_{j'} &= \frac{\alpha_{i'}}{\alpha_{j'}} \Sigma_k + \left((\beta \eta^2 - \beta - \eta^2) \frac{\alpha_k}{\alpha_{j'}} + 1 \right) \vec{a}_1^{(k)} (\vec{a}_1^{(k)})^T \end{aligned} \quad (3.18)$$

where $\{\eta, \beta\} \in (0, 1)$ are random variables defined by users. The parameter estimation based on the singular value decomposition obeys the regulations stated in Eq. (3.16) by substituting the parameter estimation equations Eq. (3.18).

3.4.3.2 Cholesky Decomposition

Again, since each covariance matrix is symmetric positive definite, the matrix Σ_k of component k can be decomposed to $\Sigma_k = L_k L_k^T$ using Cholesky decomposition, where L_k is an upper triangular matrix with positive diagonal elements. Suppose the covariance matrices of the new components i' and j' are also decomposed as $\Sigma_{i'} = L_{i'} L_{i'}^T$ and $\Sigma_{j'} = L_{j'} L_{j'}^T$ respectively, the

matrices $L_{i'}$ and $L_{j'}$ are estimated from L_k by

$$\begin{aligned} L_{i'} &= L_k \text{diag} \left(\sqrt{\beta(1-\eta^2) \frac{\alpha_{j'}}{\alpha_{i'}}}, \sqrt{\frac{\alpha_{j'}}{\alpha_{i'}}} \right) \\ L_{j'} &= L_k \text{diag} \left(\sqrt{\frac{\alpha_{i'}}{\alpha_{j'}}}, \sqrt{(1-\beta)(1-\eta^2) \frac{\alpha_{i'}}{\alpha_{j'}}} \right) \end{aligned} \quad (3.19)$$

Again, $\{\eta, \beta\} \in (0, 1)$ are random variables defined by users. The method of estimating the means $\mu_{i'}$ and $\mu_{j'}$ is the same as given in Eq. (3.18) but $\vec{a}_1^{(k)}$ is the first column vector of L_k in this case. The second split method based on Cholesky decomposition also obeys the relation constraints from Eq. (3.16).

3.4.4 Merge and Split Criteria

At each iteration of the split and merge operation, only one pair of components and only one component are selected from the given GMM and respectively subjected to the merge step and the split. In order to complete the SMEM algorithm, suitable criteria are essential for selecting the merge pair $\{i, j\}$ and the split candidate k . Since the Zhang *et al.* SMEM algorithm adopts the merge and split criteria introduced by Ueda *et al.*, the criteria proposed in [73] are described in the following.

The merge step tries to minimize the amount of components required for the fitting. Since each component is a distribution, two components are more suitable for merging if they have high correlation in the distribution. The merge criterion proposed in [73] measures the correlation between the distributions of every two components in the GMM, and the pair with highest correlation is selected as the merge pair $\{i, j\}$. The formulation of the merge criterion is defined as

$$\begin{aligned} \{i, j\} &= \arg \max_{m_1, m_2 \neq m_2} J_{\text{merge}}(m_1, m_2; \Theta) \\ &= \arg \max_{m_1, m_2 \neq m_2} \frac{(P_{m_1}(\vec{X}, \Theta))^T P_{m_2}(\vec{X}, \Theta)}{\|P_{m_1}(\vec{X}, \Theta)\| \|P_{m_2}(\vec{X}, \Theta)\|} \end{aligned} \quad (3.20)$$

where $P_{m_1}(\vec{X}, \Theta) = \{p(m_1|\vec{x}_1, \Theta), \dots, p(m_1|\vec{x}_N, \Theta)\}$ is an N-dimensional vector consisting of the posterior probability (Eq. (3.11)) for component m_1 with respect to given data set $\vec{X}$ with N elements, and $\|\cdot\|$ is the Euclidean vector norm.

In contrast, the split step tries to appropriately increase the number of the components so that the mixture model fits the data better. Hence, the split criterion measures the fitting of the components to the given data, and selects the ‘worst-fit’ component to split. For the splitting point of view, the data points ‘covered’ by the ‘worst-fit’ component should be distributed very differently from the distribution ‘described’ by the component. In [73], the actual distribution of the data points ‘covered’ by component m is governed by the local data density $f_m(\vec{x}_n; |\theta)$

that defined as

$$f_m(\vec{x}_n; |\theta) = \frac{\sum_{c=1}^N \delta(\vec{x}_n) p(m|\vec{x}_c, \Theta)}{\sum_{c=1}^N p(m|\vec{x}_c, \Theta)}$$

The split criterion proposed in [73] measures the difference between the actual distribution $f_m(\cdot)$ of the data points and the distribution $p_m(\cdot)$ of each component $m \in [1, M]$ using Kullback divergence. The component with highest difference is selected as the split candidate k . The formulation of the split criterion is defined as

$$\begin{aligned} k &= \arg \max_m J_{split}(m; \Theta) \\ &= \arg \max_m \sum_{n=1}^N f_m(\vec{x}_n; |\theta) \log \frac{f_m(\vec{x}_n; |\theta)}{p(\vec{x}_n|\theta_m)} \end{aligned} \quad (3.21)$$

3.4.5 Discussion

This section has given a review of the basic EM algorithm for estimating the parameters of GMM, and the extended SMEM that enables the EM algorithm to decide the number of components in the iterative estimation process. The basic SMEM algorithm that was firstly established by Ueda *et al.* in [73] presented the idea of split-and-merge operation by providing a simple merge method and a split method with associated merge and split criteria as well as a model selection criterion using likelihood function. Lately, in [99], Zhang *et al.* pointed out the problem of in-dependency in estimating the mean μ and covariance Σ in [73], and introduced a new merge method and two alternative split methods.

The two split methods presented by Zhang *et al.* use different matrix decomposition techniques, and both compute the means and covariances of the new components dependently based on decomposing the covariance matrix of the splitting component. However, both the Ueda *et al.* SMEM algorithm and the Zhang *et al.* SMEM algorithm are stochastic. As outlined in Eq. (3.14), the Ueda *et al.* split method relies on two variables $\{\epsilon, \epsilon'\}$. Also, as described in Section 3.4.3, the Zhang *et al.* split methods are reliant on three variables $\{\gamma, \nu, \beta\} \in (0, 1)$. All these variables are either set manually by the users (become constant) before running the process, or being set automatically and randomly during the process. However, the variables with constant values would not induce good fittings to every spot when the spot shapes are various. This means the first approach has low modelling performance and is not preferable. On the other hand, the second approach using random values can generate good fittings, but the ‘randomness’ can cause different outputs (Fig. 3.6) to the split process so that the shape model becomes un-deterministic and hence is not repeatable. Since repeatability is a key consideration in data analysis for biologist researches (so the results of repeated analysis are the same), an un-repeatable model is not suitable for this work. Although the EM process associated to the split operation can reduce the randomness by convergence, this convergence is only guaranteed to reach a local optimum but not the global optimum. As show in Fig. 3.6, since the fitting process for each spot can have multiple output (i.e. multiple local optima), the repeatability

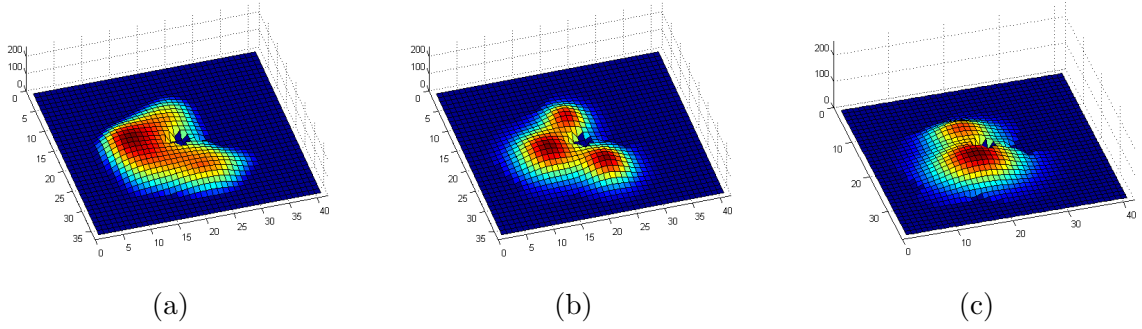


Figure 3.6: Variation in the results of stochastic split process. (a) A sample of pre-processed spot-region; (b) & (c) are two different output estimates of (a) caused by random variables in split operation.

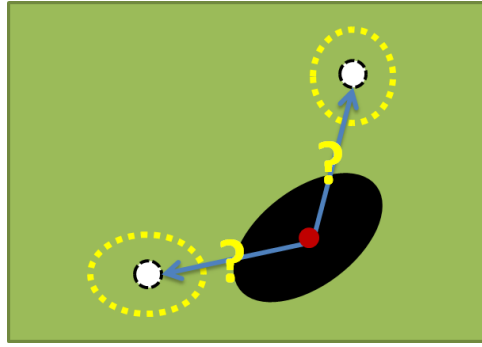


Figure 3.7: Uncertainty of split operation (This figure illustrates the fundamental problem of splitting a component (black ellipse) into two new components (yellow dash ellipse) is uncertainty of the optimal positions of the new components.)

problem of the second approach is still not solved.

In order to remove the randomness from the split algorithm and design a repeatable modelling algorithm, it is important to firstly understand the function of the variables in the process. The fundamental problem of the split operation is as shown in Fig. 3.7. The positions of the new components that are being split from a current component cannot be determined from the given data, and the number of unknown parameters is twice that of the known parameters.

Unlike data point clustering, when using GMMs to model an intensity profile, the target profile is available, and can be used to determine how well the model fits the data. Intuitively, this estimation error may be novel for finding a new split method that is simple and reliable, and at the same time is deterministic. In the algorithm introduced in this work, a new element called the ‘error distribution’ is introduced. The ‘error distribution’ is the error ‘induced’ by each individual component, and is the key to the new split algorithm leading to a new SMEM algorithm, the error-based SMEM (eSMEM) algorithm. Also, a constraint is induced from the given ‘target’ shape is injected to the basic EM algorithm in Eq. (3.12) which is then customized

for the eSMEM shape modelling algorithm.

3.5 The eSMEM Algorithm for Shape Modelling

The goal of the shape modelling in this work is to fit a GMM to the surface profile of each particle-cluster. The GMMs typically represent PDFs and hence the integral of a GMM over all possible values of dependent variables is 1. However, the surface profile of a particle-cluster is actually the intensity appearance $\mathbf{I}$ of that object in the image where the value is typically in the range $[0, 255]$. Direct use of $\mathbf{I}$ in Eq. (3.1) can cause failure in the estimation process due to the range of intensity values. In order to correct the surface profile intensities without distorting the shape of the particle-cluster, the intensity appearance $\mathbf{I}$ of the modelling object is first normalized according to

$$I_{norm}(\vec{x}) = \frac{I(\vec{x})}{\sum_{n=1}^N I(\vec{x}_n)} \quad (3.22)$$

where $\vec{x} = (h, k)$ is a pixel site in this case. Consequently, the GMM in Eq. (3.1) then becomes

$$I_{norm}(\vec{x}) = \sum_{n=1}^N \alpha_n N(\vec{x} | \vec{\mu}_n, \Sigma_n). \quad (3.23)$$

The shape of a particle-cluster, i.e. the normalized ‘intensity’ I_{norm} is then subjected to the SMEM modelling process to estimate the parameters $\{\theta_1, \dots, \theta_m, \dots, \theta_M\} \in \Theta$ of the GMM model, where $\{\alpha_m, \mu_m, \Sigma_m\} \in \theta_m$ is the distribution parameters of the m^{th} component of the GMM which has M components. The customization of the EM algorithm required for shape modelling is described in the next section.

3.5.1 EM Algorithm for Shape Modelling

Traditionally, the EM algorithm for GMM for data points clustering fits the model to the distribution of the data points in the domain of the data set. The idea is that a GMM as an PDF is fitted to the distribution of the data points and hence represents the probability of detecting a data point at a certain coordinate in the domain. For shape modelling, the normalized intensity at a pixel site is synonymous with probability.

For the purpose of shape modelling, the ‘shape’ of a particle-cluster is the brightness profile of the merged cluster object, and this apparent brightness is a combination of the luminance of each single particle in the cluster. Since the point-spread-function [41, 97] is the same for all the particles in the same image, and the luminance of each particle is approximated by a Gaussian distribution, the apparent brightness of a particle cluster is then approximated by a mixture of the Gaussian distributions, i.e. the GMM. From the GMM for data points clustering to shape modelling, the variable $\vec{x}$ becomes the pixel position (h, k) , and the frequency of realization at $\vec{x}$ becomes the normalized ‘intensity’ value at (h, k) . In other words, the probability of a variable $\vec{x}$ in a GMM is now mapped to the surface brightness of a particle-cluster at pixel site (h, k) .

The ultimate goal of the shape modelling is to find the number, positions and sizes of the particles in a given cluster. These factors can be parameterized by the parameters Θ of the GMM fitted to the cluster. The number of the particles can be mapped to the model order M of mixture components in the model, the position and size of each particle can be represented by the mean μ and the covariance Σ respectively of the corresponding component in the model.

In *Section 3.3*, the posterior probability of the hidden variable y_m in Eq. (3.11) is a the probability of a particular data point $\vec{x}_n$ being generated by a particular mixture component of the GMM. Based on this assertion, the hidden variable y can intuitively be considered as the ‘label’ of the mixture component, where the probability for y with value m at coordinate $\vec{x}_n$, is the probability of a data point at $\vec{x}_n$ that is generated by the component m . For shape modelling, since the intensity at each pixel is assumed to be the sum of all components at that pixel, the ‘probability’ of m is then rephrased as the ‘proportion’ of m that indicates how much of the intensity at pixel $\vec{x}_n$ is ‘generated’ by the component m . This means, while α is the ‘global’ mixing weight for mixture components over the whole modelling spot, the probabilistic measure of $y \in [1, M]$ can be considered as the ‘local’ mixing weights of the mixture components at each pixel site and denoted as $W = \{\mathbf{w}_1, \dots, \mathbf{w}_M\}$, where $w_m(x_n)$ is the local mixing weight of component m among all the components at pixel site x_n .

Clearly, for shape modelling, the fitting operation should consider the associated density value of each input coordinate (pixel site) rather than the frequency of realization of the coordinates. Since the conventional EM algorithm is based on the frequency of realization, it is modified to suit shape modelling by injecting the intensity density into the parameter update equations in Eq. (3.12) as the ‘weights’ associated with the input variable $\vec{x}$. Therefore, given a set of parameter estimates $\Theta' = \{\theta'_1, \dots, \theta'_M\}$ for the GMM with M components, where $\{\alpha_m, \vec{\mu}_m, \Sigma_m\} \in \theta_m$ is distribution parameters for the m^{th} component with the Gaussian density function $N_M(\vec{x}|\theta'_m)$, the estimation equations for component m are modified from Eq. (3.12) and are expressed as

$$\begin{aligned}
 w_m(\vec{x}_n) &= p(m|\vec{x}_n, \Theta') = \frac{\alpha_m^{(i)} N_m(\vec{x}_n|\theta_m^{(i)})}{\sum_{l=1}^M \alpha_l^{(i)} N_l(\vec{x}_n|\theta_l^{(i)})} \\
 \alpha_m^{(i+1)} &= \frac{1}{N} \sum_{n=1}^N w_m(\vec{x}_n) \\
 \mu_m^{(i+1)} &= \frac{\sum_{n=1}^N \vec{x}_n w_m(\vec{x}_n) I_{norm}(\vec{x}_n)}{\sum_{n=1}^N w_m(\vec{x}_n) I_{norm}(\vec{x}_n)} \\
 \Sigma_m^{(i+1)} &= \frac{\sum_{n=1}^N w_m(\vec{x}_n) I_{norm}(\vec{x}_n) (\vec{x}_n - \mu_m^{(i+1)}) (\vec{x}_n - \mu_m^{(i+1)})^T}{\sum_{n=1}^N w_m(\vec{x}_n) I_{norm}(\vec{x}_n)}
 \end{aligned} \tag{3.24}$$

To complete the EM algorithm for shape modelling, an initialization step is designed for the algorithm to estimate the initial values $\Theta^{(0)}$ for Θ , and is described in the next section. Also, the termination scheme in conventional EM algorithm using likelihood probability is replaced by a

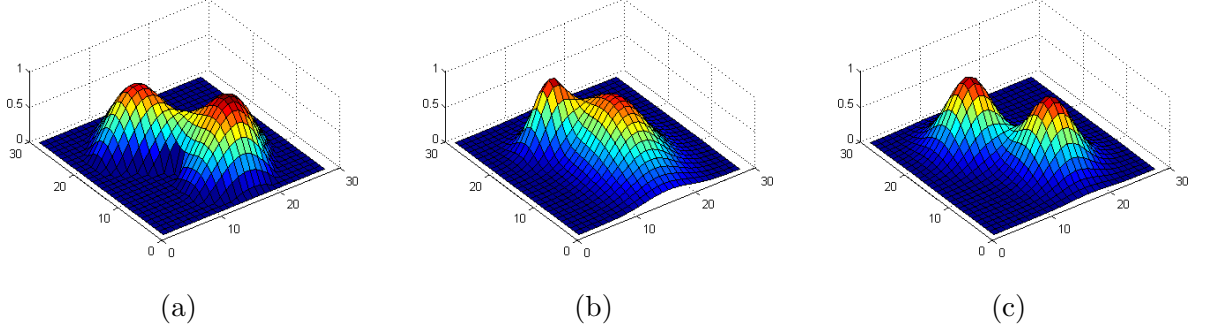


Figure 3.8: An example of EM shape modelling. (a) the normalized intensity of a spot patch; (b) the initial estimate of the spot shape; (c) The modelling process terminates when the estimation meets a defined constraint.

new criterion based on estimation error. Since the ‘target’ shape $\mathbf{I}_{norm}$ for shape modelling is present, the termination scheme for the iterative process is intuitively based on the error-signal-ratio τ of the estimated model $\mathbf{I}_{EM}$ which is calculated by

$$\tau = \frac{\sum_{n=1}^N |I_{EM}(x_n) - I_{norm}(x_n)|}{\sum_{n=1}^N I_{norm}(x_n)} = \sum_{n=1}^N |I_{EM}(x_n) - I_{norm}(x_n)| \quad (3.25)$$

The termination criterion for shape modelling is defined by the rate-of-convergence of τ in the iterative process as follows

$$\frac{|\tau^{(i+1)} - \tau^{(i)}|}{\tau^{(i)}} < T \quad (3.26)$$

where T is a user defined threshold and is set to 0.1 in this work. An example of the customized EM algorithm for shape modelling is given in Fig. 3.8.

3.5.2 Initialization of the GMM

In SMEM shape modelling, an initial setting $\Theta^{(0)}$ of the GMM parameters Θ is required to start off the iterative process for a given spot patch. The initial setting includes the number $M^{(0)}$ of mixture components and the distribution parameters $\{\alpha^{(0)}, \mu^{(0)}, \Sigma^{(0)}\}$ of the Gaussian components. Firstly, the initial number $M^{(0)}$ is set to the number of peaks, i.e. local maxima, in the intensity profile of given spot patch. The initial mixing weights $\alpha^{(0)}$ are set to be the same for all mixture components, and since the mixing weights are constrained by $\sum_{m=1}^M \alpha_m = 1$, $\alpha^{(0)}$ for each component is set to $\frac{1}{M^{(0)}}$. For a 2D Gaussian distribution, the mean μ is located at the mode of the distribution, consequently the default means $\mu^{(0)}$ of Gaussian components are set to the positions of the peaks $\vec{X}_{max} = \{\vec{x}_{max,1}, \dots, \vec{x}_{max,M0}\}$ in the given spot patch. Finally, the initial covariance $\Sigma^{(0)}$ of each Gaussian component is estimated, as shown in Fig. 3.9, based on the selected pixels $\vec{X}_s$ around the corresponding mean $\mu^{(0)}$. For an initial covariance $\Sigma_m^{(0)}$, the pixels $\vec{X}_{s,m}$ participate in the estimation are selected from a circular region centred at $\mu_m^{(0)}$ with a radius equal to the shortest distance from $\mu_m^{(0)}$ to the spot boundary. Since the idea is to get

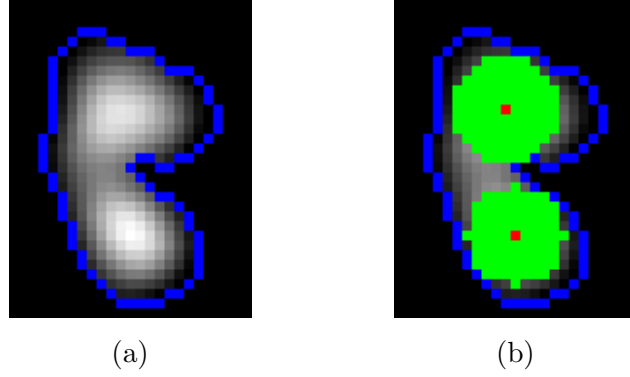


Figure 3.9: An example of pixel selection for initial covariance estimation. (a) the ‘blue’ boundary of the spot is defined by an edge detector, e.g. difference of Gaussians (DoG); (b) the pixels used for estimating the initial covariances are selected from the areas (in green) that are centred at the corresponding means (in red) of the two mixture components. the radius of each area is the shortest distance between the corresponding mean and the spot boundary.

a rough estimate of the model parameters, the exact boundary of the region is not of interest. Instead, the boundary is defined by a standard edge detector, like difference of Gaussians (DoG, $\sigma_1 = 1$ & $\sigma_2 = 1.5$ in this work). This is because the resulting region can theoretically provide enough of the profile of the region to enable accurate estimation of the GMM. Consequently, the parameter initialization of component m is defined as

$$\begin{aligned}
 \alpha_m^{(0)} &= \frac{1}{M^{(0)}} \\
 \mu_m^{(0)} &= \vec{x}_{\max, m} \\
 \mu_m^{(0)} &= \frac{\sum_{\vec{x}_s \in \vec{X}_{S, m}} I_{norm}(\vec{x}_s) (\vec{x}_s - \mu_m^{(0)}) (\vec{x}_s - \mu_m^{(0)})^T}{\sum_{\vec{x}_s \in \vec{X}_{S, m}} I_{norm}(\vec{x}_s)}
 \end{aligned} \tag{3.27}$$

As shown in Fig. 3.13, the initial setting $\Theta^{(0)}$ is subjected to the modified EM algorithm in Eq. (3.24), and the converged result is denoted as Θ^* . Θ^* is then subjected to the iterative split-and-merge operation in order to optimize the model order M of the GMM. As discussed earlier, the merge and split steps, and the criteria for selecting the merge candidates and split candidate are the core of the operation. The merge method and a novel split method as well as the new merge and split criteria that customized for shape modelling are presented in the following sections.

3.5.3 Split and Merge Operation

Recall that the purpose of the split-and-merge operation is to enable the EM algorithm to estimate the number M of components. In this work, as mentioned in Section 3.1, because of the problem of 2D projection of 3D structure, the cluster spots, as shown in Fig. 3.6(a), have merged intensity profile. The fundamental problem for cluster analysis is that the number and

shapes of the candidate particles in each cluster spot are very difficult to determine directly from the ‘shape’ of the spot. The initial setting $M^{(0)}$ for the number M , as described in the previous section, is defined simply by the number of peaks in the intensity profile of the given spot. In most cases, $M^{(0)}$ is not the optimal number of components for fitting the given cluster spots. Therefore, a split-and-merge operation is employed in the EM shape modelling process, and the error-based SMEM (eSMEM) algorithm presented in the following section is customized for shape modelling.

For the merge step, because the parameter estimation for the new component from the two merging components is an overdetermined process, this operation is simple and deterministic. Subsequently, the merge step of the eSMEM algorithm adopts the merge method presented by Zhang *et al.* in [99] which has been reviewed in *Section 3.4.2* and has been adopted in several recent SMEM modelling applications [60, 63]. For the split step, as discussed earlier in *Section 3.4.5*, the split methods designed by Zhang *et al.* in [99] are stochastic and hence fails the requirement for repeatability for shape modelling in this work. Therefore, in the following sections, a determined split method is proposed for shape modelling. It is based on employing the fact that the target density I_{norm} is known in this problem.

3.5.4 Error Distribution

Recall that the estimation error $\mathbf{E}$ is defined as the difference between the model $\mathbf{I}_{EM}$ and the target shape $\mathbf{I}_{norm}$ as follows

$$E(\vec{x}_n) = I_{EM}(\vec{x}_n) - I_{norm}(\vec{x}_n) \quad (3.28)$$

For shape modelling, the ultimate goal is to minimize the fitting error $\mathbf{E}$ by optimizing the GMM. Since the split operation only operates on one mixture component at a time without affecting the ‘status’ (the parameters) of the other components, the goal of the split operation is to minimize the estimation error induced by the component being split regardless of the error induced by the other components. However, the estimation error that ‘belongs’ to a particular mixture component is different from $\mathbf{E}$ which results from the combined shape of all the mixture components. In order to distinguish it from $\mathbf{E}$, the estimation error that ‘belongs’ to a single component is named the ‘error distribution’, and denoted as $\mathbf{e}_m$ for component m . Therefore, an error distribution $\mathbf{e}_m$ is a distribution of estimation error induced by the corresponding component m at every pixel site in the given spot patch, and is a key idea to create a deterministic split algorithm. However, calculating the error distribution is not obvious as $\mathbf{E}$ is only directly connected to the GMM ($\mathbf{I}_{EM}$) and not each individual component. The fact that $\mathbf{E}$ is an overall estimation error induced by the combined model is equivalent to $\mathbf{E}$ being the sum of the error distribution of each component. Unfortunately, the reverse process of estimating a candidate error distribution from the combined error $\mathbf{E}$ is also an under-determined problem.

A simple and direct solution to this problem would be to divide $\mathbf{E}$ by the mixing weights α of the components. However, $\mathbf{E}$ is a distribution of pixel-wide error covering the entire spot patch, and since the mixture components are distributed at different locations in the patch, the same partitioning ratio for the error at all pixel sites is inappropriate. In fact, a component should share more error in the region closest to its mean. This means the error partitioning ratio is more appropriate to be defined at the pixel-scale. However, since the actual number M of the particles in the spot is not determined, finding the exact partitioning ratio at each pixel becomes an under-determined problem. To generate more constraints for this problem, an assumption is made that a component that contributes more to the overall intensity at a particular pixel site is assumed to generate a larger portion of the estimation error, and conversely the component that contributes less to the intensity generates proportionally less error. Therefore, the component error (the error distribution e_m) is proportional to the local mixing weight w of the GMM which is previously defined in Eq. (3.24) and is given by

$$e_m(\vec{x}_n) = w_m(\vec{x}_n)E(\vec{x}_n). \quad (3.29)$$

The detail of the new split method that conducted by the error distribution $\mathbf{e}$ is described in the following section. Furthermore, the split and merge criteria for the eSMEM shape modelling algorithm are also based the error distribution $\mathbf{e}$, and is discussed in *Section 3.5.6*.

3.5.5 Error-based Split Method

As discussed in *Section 3.4.3* the split operation divides a selected mixture component k into two new components $\{i', j'\}$. For GMM, the key problem of the underdetermined process is the uncertainty of positioning the new components in the spot patch. Since the ‘position’ of a 2D Gaussian distribution is indicated by the peak (the mean μ) of the distribution, accordingly finding the positions of i' and j' is equivalent to estimating the means $\mu_{i'}$ and $\mu_{j'}$ from μ_k of k . This process of finding the new mean values is referred to as mean-splitting. The estimation of the new means is difficult for the conventional SMEM algorithm due to the lack of information apart from the coordinates of the data points. On the contrary, for shape modelling, the error distribution e_k of the selected component k can help to locate the positions of the new components. In this work, the means $\mu_{i'}$ and $\mu_{j'}$ are located between the maximum and different respective minima in the profile of the error distribution e_k as shown in Fig. 3.10.

In order to explain why the means are located in this way, an illustration using a simpler 1D model is firstly shown in Fig. 3.11. In this example, a synthetic sample distribution is generated by combining two 1D Gaussian densities (blue dash curves) i' and j' respectively. This synthetic distribution (blue solid curve) is the ‘target shape’ of the shape modelling operation and the sum of curves i' and j' is the optimal output of modelling. The ‘target shape’ is fitted with a GMM (green curve) with only one component k . The difference between the GMM and the ‘target shape’ is represented by the red curve and is the estimation error of the model. Since the model has only one component k , the estimation error is also the error distribution of k . To

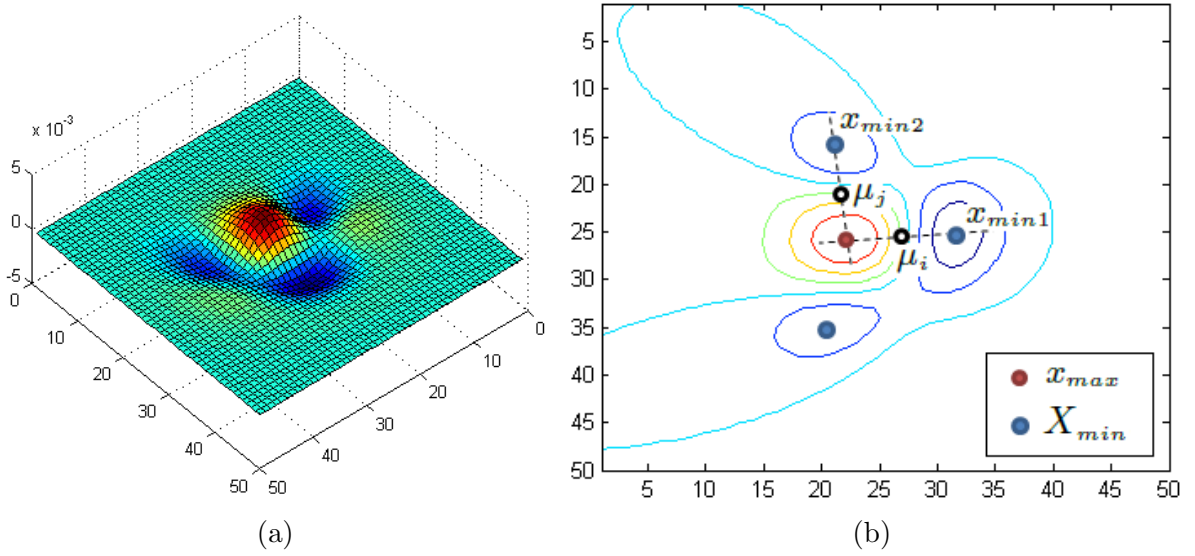


Figure 3.10: Error distribution. (a) shows the profile of an error distribution and (b) is the contour of the distribution shown in (a).

achieve a better modelling result, the GMM should split k into two new components, and the optimal output of the split is i' and j' .

The EM algorithm can optimize the new parameters of the GMM after the split, but the key is to find the right positions of the new components as the initial input to the EM operation, where the position of a component is indicated by the mean μ , i.e. the peak, of that component. Again, this problem is difficult to solve in the case of data clustering. However, in shape modelling, the existence of the error distribution can provide extra information for finding the positions of the new components. As highlighted by the black dash lines in Fig. 3.11, the peak of i' lies correspondingly between the maximum and the ‘left’ local minimum of the error distribution of k . Similarly, the peak of j' is between the maximum and the ‘right’ local minimum. This means, in the 1D model, the two optimal split positions sit between the maximum and the different minima respectively in the error distribution of the current component being split.

The splitting metric of the error distribution remains valid for the 2D GMM. To illustrate this, a synthetic 2D distribution is generated by combining two designed Gaussian densities, and is fitted by a 2D GMM with only one component. The 2D model of the splitting metric of error distribution is illustrated in the contour representation in Fig. 3.12(a). The target distribution at the middle level is formed by combining the two designed Gaussian densities i' and j' . A 2D GMM shown at the second bottom level is fitted to the target distribution with a single component k , and the difference between the GMM and the ‘target’, i.e. the error distribution, is shown at the bottom of Fig. 3.12(a). Again, the estimation error is also the error distribution of k , and in order to have a better fitting to the target shape, the GMM needs two components by splitting the current single component k . Once more, the Gaussian densities i' and j' are the optimal output of the split process on k . The black dash lines with the red crosses highlight that

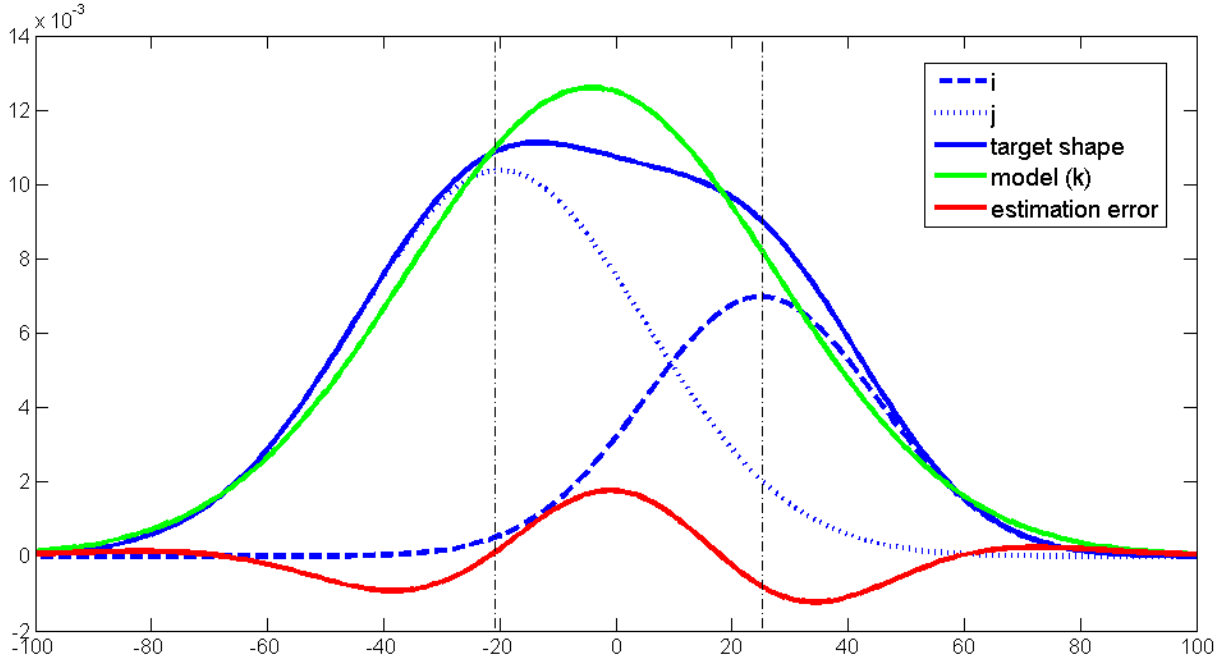


Figure 3.11: Using the error distribution for mean-splitting in 1D (*This figure uses an artificial 1D model to illustrate how the error distribution is used to locate the means of the new components in the split operation. A distribution (blue solid curve) is generated by combining two designed Gaussian densities i' and j' (blue dash curves). This synthetic distribution is the ‘target shape’ for modelling by a GMM with only one component k (green curve). The difference between the GMM and the ‘target’ is the represented by the red curve, which is also the error distribution of k . The black dash lines highlight that the means of the optimal split results i' and j' are located between the maximum and the different minima in the error distribution of k*)

the means of i' and j' , i.e. the peaks of the distributions, are located in the regions between the maximum and different respective minima in the error distribution of k . This means that while the component locations of the target shape are unknown, the optimal new components being split from a current component would be located in the regions that lie between the maximum and minima in the error distribution of that current component.

3.5.5.1 Split on Real Data

The error distributions (as shown in Fig. 3.10(a)) of the GMMs fitted to real data have similar characteristics to the 2D example in Fig. 3.12(b), which has a maximum and a small number of minima. Based on the synthetic models shown above, the optimal new components of the split operation for real data are assumed to be located in the regions between the maximum and two different minima in the error distribution of the selected component for splitting. However, in contrast with the synthetic models, the error distribution of a component in a GMM is more

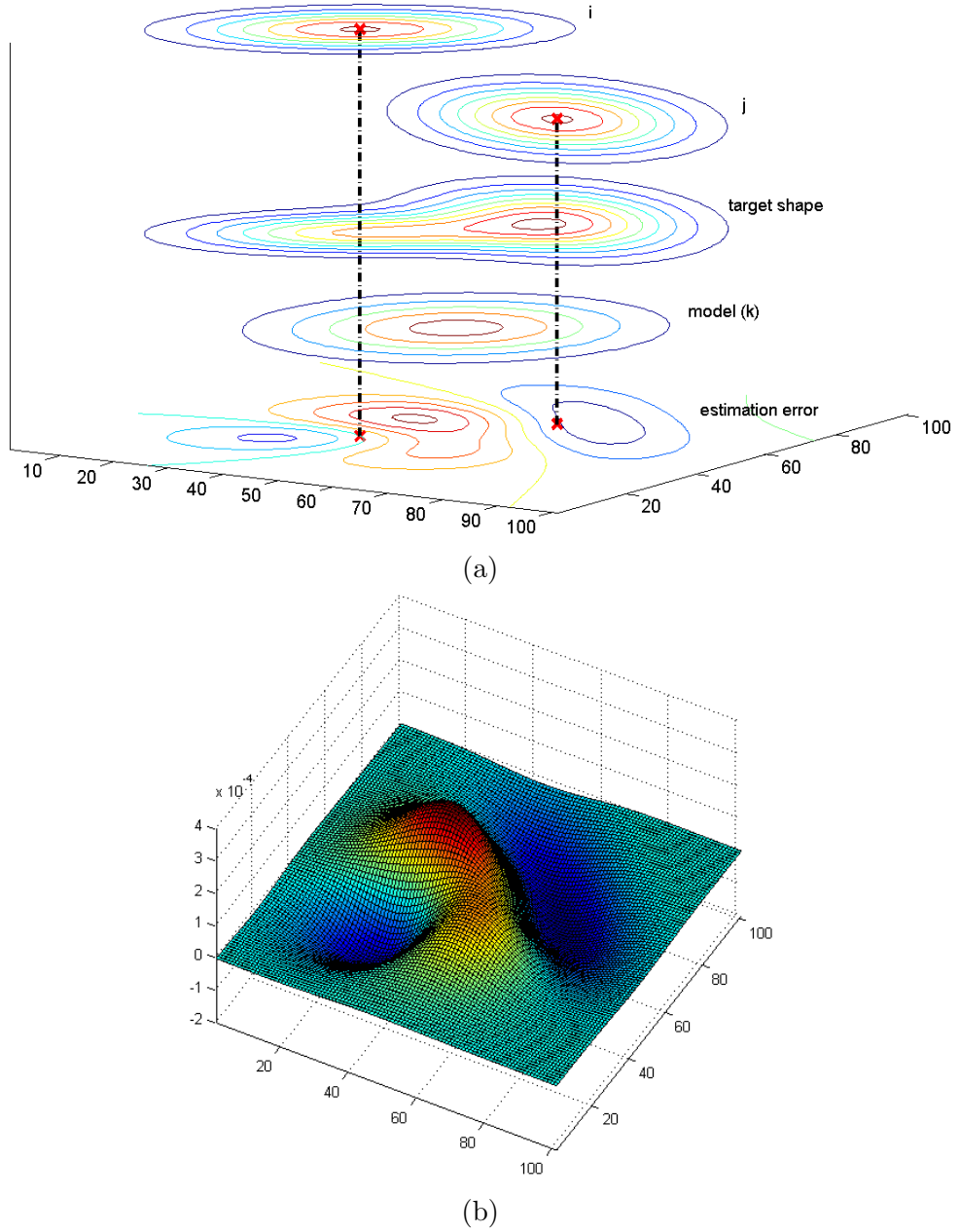


Figure 3.12: Using the error distribution for mean-splitting in 2D ((a) illustrates the split metric of error distribution with a 2D model in the contour representation. The contour in the middle level represents the synthetic target distribution that is generated by combining the two Gaussian densities i' and j' shown by the contours on the top two levels. The ‘target’ distribution is fitted by a GMM with only one component k shown at the second bottom level. The difference between the GMM and the ‘target’, which is also the error distribution of k is given at the bottom. The black dash lines with the red crosses show that the two optimal positions of the splitting, i.e. i' and j' the candidates of the ‘target’, are located in the regions that are between the maximum and different minima in the error distribution of k . (b) shows the profile of error distribution which is corresponding to the contour representation at the bottom of (a).)

complex. Recall that the error distribution $\mathbf{e}_k$ is assumed to be the error induced by component k and is divided from the overall error $\mathbf{E}$ by the local mixing weight $\mathbf{w}_k$. The assumption is more likely to be true in the corresponding region close to the mean $\vec{\mu}_k$ of the k , but this does not mean $\mathbf{e}_k$ is independent from the other components. In fact, $\mathbf{e}_k$ may share the error predominantly induced by some other components which should be split or merged, especially in the regions further away from $\vec{\mu}_k$. Consequently, the error distribution $\mathbf{e}$ for real data, as shown in Fig. 3.10(a), can have more or less than two minima in the distribution.

Since the split operation only considers a single selected component at a time, the split method is only based on the corresponding error distribution for that component, and the key is to select the minima from the error distribution used to decide the locations for the means of the new components. Note that a less well fitted component would cause larger estimation error and hence have a higher maximum and lower minima in the corresponding error distribution. Consequently, in the case of more than two minima (X_{min}) in the given error distribution, the minima being considered in the split process would be the two minima with the lowest values ($\{\vec{x}_{min1}, \vec{x}_{min2}\}$) respectively. The new means $\{\vec{\mu}_i, \vec{\mu}_j\}$ are located half way between the maximum ($\vec{x}_{max}$) and the respective minima ($\{\vec{x}_{min1}, \vec{x}_{min2}\}$). This also applies to the situation with exactly two minima. For the case with only one minimum ($\vec{x}_{min}$), $\vec{x}_{max}$ would be one of the new means is set by $\vec{x}_{max}$ and the other one is set to the mid-point of $\vec{x}_{min}$ and $\vec{x}_{max}$.

As shown by the synthetic models in Fig. 3.11 & Fig. 3.12, the candidate component with higher mixing weight in the ‘target’ model is correspondingly closer to the minima with higher value, i.e. lower absolute value in the error distribution, and oppositely, the one with lower mixing weight is close to the minima with higher absolute value. In the split operation, since the new mixing weights $\{\alpha_i, \alpha_j\}$ are obtained from the original mixing weight α_k by a particular partitioning ratio as follows

$$\alpha_i' = \frac{|e_k(\vec{x}_{min2})|}{|e_k(\vec{x}_{min1})| + |e_k(\vec{x}_{min2})|} \quad (3.30)$$

$$\alpha_j' = \frac{|e_k(\vec{x}_{min1})|}{|e_k(\vec{x}_{min1})| + |e_k(\vec{x}_{min2})|} \quad (3.31)$$

For only one minimum exists, $\{\alpha_i', \alpha_j'\}$ divide α_k equally, i.e. $\frac{\alpha_k}{2}$.

The new covariance matrices $\{\Sigma_i, \Sigma_j\}$ are estimated by the estimation equation in Eq. (3.24) using the new means $\{\vec{\mu}_i, \vec{\mu}_j\}$ and new local mixing weights $\{\mathbf{w}_i, \mathbf{w}_j\}$ which are derived from $\mathbf{w}_k$ with the new mixing weight $\{\alpha_i, \alpha_j\}$. The overall algorithm for the error-based split method is given in *Algorithm 1* with the help of illustration in Fig. 3.10(b).

Accordingly, based on the error distribution $\mathbf{e}_k$ of the selected component k having a maximum $\vec{x}_{max}$ and a small number N_{min} of minima X_{min} , the new split method for shape modelling is introduced in *Algorithm 1* with the help of illustration in Fig. 3.10(b). To complete the split operation, a split criterion is required for selecting the split component k . However, since the goal of shape modelling is to minimize the estimation error, the split criterion for the eSMEM algorithm would be based on the error distribution which is different to the split criterion for

Algorithm 1 Error-based Split Method

```

if  $N_{min} = 1$  then
     $\alpha_{i_l} = \alpha_{j_l} = \frac{1}{2}\alpha_k$ 
     $\vec{\mu}_{i_l} = \vec{x}_{max}$ 
     $\vec{\mu}_{j_l} = \frac{1}{2}(\vec{x}_{max} + \vec{x}_{min})$ 
else
     $\vec{x}_{min1} = \min_{\vec{x}}\{e_k(\vec{x}); \vec{x} \in X_{min}\}$ 
     $\vec{x}_{min2} = \min_{\vec{x}}\{e_k(\vec{x}); \vec{x} \in X_{min}, \vec{x} \neq \vec{x}_{min1}\}$ 
     $\alpha_{i_l} = \frac{|e_k(\vec{x}_{min2})|}{|e_k(\vec{x}_{min1})| + |e_k(\vec{x}_{min2})|}\alpha_k$ 
     $\alpha_{j_l} = \frac{|e_k(\vec{x}_{min1})|}{|e_k(\vec{x}_{min1})| + |e_k(\vec{x}_{min2})|}\alpha_k$ 
     $\vec{\mu}_{i_l} = \frac{1}{2}(\vec{x}_{max} + \vec{x}_{min1})$ 
     $\vec{\mu}_{j_l} = \frac{1}{2}(\vec{x}_{max} + \vec{x}_{min2})$ 
end if
 $\mathbf{w}_{i_l} = \frac{\alpha_{i_l}}{\alpha_k}\mathbf{w}_k$ 
 $\mathbf{w}_{j_l} = \frac{\alpha_{j_l}}{\alpha_k}\mathbf{w}_k$ 
 $\Sigma_{i_l} = \frac{\sum_{n=1}^N w_i(\vec{x}_n) I_{norm}(\vec{x}_n)(\vec{x}_n - \vec{\mu}_{i_l})(\vec{x}_n - \vec{\mu}_{i_l})^T}{\sum_{n=1}^N w_i(\vec{x}_n) I_{norm}(\vec{x}_n)}$ 
 $\Sigma_{j_l} = \frac{\sum_{n=1}^N w_j(\vec{x}_n) I_{norm}(\vec{x}_n)(\vec{x}_n - \vec{\mu}_{j_l})(\vec{x}_n - \vec{\mu}_{j_l})^T}{\sum_{n=1}^N w_j(\vec{x}_n) I_{norm}(\vec{x}_n)}$ 

```

data clustering. This is also the case th the merge criterion for sorting the merge candidates. The detail of the error-based split and merge criteria is given in the next section.

3.5.6 Split and Merge Criteria for shape modelling

The split step and the merge step require one component and a pair of components at each iteration of the eSMEM. Again, split and merge criteria are required for ordering the mixture components and to select the split component and the merge component pair. The conventional criteria in [73] have been reviewed in *Section 3.4.4*, however these approaches are not suitable for shape modelling as the evaluation methods are not related to the target shape information. Instead, the new split and merge criteria for shape modelling are derived from the error distribution.

First, the split criterion selects the ‘worst fit’ component as input to the split operation. The conventional approach uses the Kullback-Leibler divergence to measure the distance between the local data density and the Gaussian density of each component. In shape modelling, the ‘fit’ can be determined by the estimation error $\mathbf{E}$, and the ‘worst fit’ component is the component causing the largest portion of the error. While the error distribution of each component has been estimated during the split operation, and is assumed to be the pixel-scale error that ‘belongs’ to the corresponding component, the ‘fit’ of a component can then be measured by the spatial sum of the absolute values of the error distribution. Consequently, the split criterion for shape

modelling is defined by selecting the component with the highest sum of the absolute error as follows

$$m_{split} = \arg \max_m \left(\sum_{n=1}^N |e_m(x_n)| \right) \quad (3.32)$$

where m_{split} is the ‘worst fit’ component with highest sum of the absolute error distribution.

In contrast, the purpose of the merge criterion is to select a pair of merge candidates $m_{merge}(i, j)$ from the GMM so that the merge process for shape modelling can reduce the number of the mixture components in the model without increasing the fitting error, and also remove false components caused by noise. Since the ‘shape’ of a GMM is dominated by the ‘large’ mixture components where the ‘size’ is determined by the associated mixing weights α , merging the ‘large’ components should be avoided in order to prevent sudden change in the ‘shape’. Instead, merging a ‘small’ component into a bigger one is preferable, and hence one of the merging candidates, $m_{merge}(i)$ is set by the component with the lowest mixing weight α .

The major problem in developing the merge criterion is finding the other component $m_{merge}(j)$ to pair up with $m_{merge}(i)$ for the merging process. The approach in the conventional SMEM algorithm is, as described in *Section 3.4.4*, defined by measuring the correlation coefficient of the posterior probability, i.e. the ‘local mixing weight’ w , between every pair of the mixture components. In other words, the merge criterion is a measurement of how much overlapping of the ‘local mixing weight’ distributions between any two components of the model. Again, this method is appropriate for clustering since the more overlapping between two components the closer they are, and more preferable to be merged so that it becomes one cluster. However, this merge criterion does not suit the shape modelling in this work as the 3D protein particles at different depths with respect to the scanning can be projected as overlapped objects in the cluster spots on the resulting 2D image. This means that the mixture components of the GMM that have large overlapping between the distributions can actually correspond to individual particles and hence should not be merged.

It is difficult to determine whether two components belong to one single particle or two different particles simply based on the distribution correlation. Instead, it is observed that when two mixture components have high distribution correlation but correspond to different particles, the ‘fit’ of the two components is high and hence the estimation error $\mathbf{E}$ in the ‘overlapping area’ of the two distributions is low. Conversely, the components that belong to a single particle have less ‘fit’ to the target shape and hence cause higher estimation error $\mathbf{E}$ between the distributions. Consequently, the merge criterion can be based on $\mathbf{E}$ in the region between every pair of mixture components. The pair having the highest error in the ‘overlapped area’ between the distributions is selected as the candidates for merging. In order to measure $\mathbf{E}$ in the ‘overlapping area’ of any two components, the estimation error $\mathbf{E}$ is weighted using the conventional coefficient correlation measure and the new merge criterion is therefore defined as

follows

$$\begin{aligned} m_{merge}(i) &= \arg \min_i (\alpha_i) \\ m_{merge}(j) &= \arg \max_j \left(\sum_{n=1}^N w_i(x_n) w_j(x_n) (E(x_n))^2 \right) \quad (i \neq j) \end{aligned} \quad (3.33)$$

As shown above, $m_{merge}(j)$ is assigned by the component that has the highest correlation with $m_{merge}(i)$ so that the two merge candidates have large overlapping in the distributions. And since the correlation is weighted by the set of pixel-scale estimation error $\mathbf{E}$, a bias is introduced for candidates that are less well fitted to the target shape. Interestingly, the correlation of the local mixing weights $\mathbf{w}$ of two mixture components weighted by the square of error $\mathbf{E}^2$ to avoid the cancellation between the positive and negative errors. This is equivalent to the correlation of the error distributions $\mathbf{e}$ of those two components.

$$\begin{aligned} m_{merge}(j) &= \arg \max_j \left(\sum_{n=1}^N w_i(x_n) w_j(x_n) [E(x_n)]^2 \right) \\ &= \arg \max_j \left(\sum_{n=1}^N w_i(x_n) E(x_n) w_j(x_n) E(x_n) \right) \\ &= \arg \max_j \left(\sum_{n=1}^N e_i(x_n) e_j(x_n) \right) \end{aligned}$$

Accordingly, both the new split and merge criteria for shape modelling are based on the error distribution $\mathbf{e}$ and hence are closely related to the target shape information.

3.5.7 eSMEM Modeling Process

A new split-and-merge method based on the estimation error ($\mathbf{E}$) has been introduced for shape modelling, and it is called the error-based SMEM (eSMEM) algorithm. However, in order to prevent the GMM having un-necessarily exceeded the number of components for fitting the target shape, and to increase the chance of achieving the global optimum for the algorithm, a modification is applied to the arrangement of the procedures of the conventional SMEM process. To improve convergence, a third estimation candidate is introduced to the split-and-merge operation which is derived based on a split-merge step. This means, in each iteration of the operation, there are three steps: the split step, the merge step and a split-merge step. The split-merge step applies a merge to the parameter set resulted from splitting.

The overall structure of the eSMEM algorithm is shown in Fig. 3.13. In the split-and-merge operation of the eSMEM algorithm, a set of parameters Θ^* is given to the merge step and the split step and the respective resultant parameters $\{\Theta'_{merge}, \Theta'_{split}\}$ are optimized to $\{\Theta^*_{merge}, \Theta^*_{split}\}$ by the modified EM algorithm. A merge step is applied to Θ^*_{split} to generate a new set of parameters Θ'_{sm} for the split-merge step and then optimized to Θ^*_{sm} through the EM process. Consequently, there are three new sets of parameters $\{\Theta^*_{merge}, \Theta^*_{split}, \Theta^*_{sm}\}$ with the corresponding error to signal

ratios $\{\tau_{merge}^*, \tau_{split}^*, \tau_{sm}^*\}$ at each iteration of the eSMEM. A choice must be made as to which set is the best. This decision is made by comparing the total absolute errors of the three sets of parameters with the error to signal ratio τ^* of the current parameters Θ^* .

Since a model with more components always fit the data better, the results of the three steps are associated with different priorities to prevent over-fitting. The goal of the decision process is to favor solutions that give fewer numbers of components in the model. Therefore, the merge operation Θ_{merge}^* is prioritized to reduce the number of components and the split operation Θ_{split}^* is given the least priority as it increases the number of components. The selected parameters are then assigned to Θ^* for the next iteration of the split-and-merge process. The termination scheme of the iterative split-and-merge process is similar to the criterion for EM algorithm in Eq. (3.26) defined by thresholding the rate-of-convergence of the error to signal ratio τ^* with threshold T_{sm} as follows

$$\frac{|\tau^{*(i+1)} - \tau^{*(i)}|}{\tau^{*(i)}} < T_{sm} \quad (3.34)$$

Consequently, as shown in Fig. 3.13, the procedure of the eSMEM shape modelling process is as follows:

1. The mixture parameters are initialized to Θ^0 using the method described in *Section 3.5.2*.
2. Run EM estimation with Θ^0 to yield parameters Θ^* .
3. Perform a merge-step and a split-step on Θ^* simultaneously and yield parameters Θ'_{merge} and Θ'_{split} respectively.
4. Run EM estimation with Θ'_{merge} and Θ'_{split} to yield parameters Θ_{merge}^* and Θ_{split}^* respectively.
5. Perform a merge-step on Θ_{split}^* and yield parameters Θ'_{sm} .
6. Run EM estimation with Θ'_{sm} to yield parameters Θ_{sm}^* .
7. Compare $\{\Theta_{sm}^*, \Theta_{merge}^*, \Theta_{split}^*\}$ to Θ^* by comparing the corresponding error to signal ratio $\{\tau_{sm}^*, \tau_{merge}^*, \tau_{split}^*\}$ and τ^* . If $\tau_{merge}^* < \tau^*$ then set $\{\Theta^* \leftarrow \Theta_{merge}^*, \tau^* \leftarrow \tau_{merge}^*\}$. Otherwise, if $\tau_{sm}^* < \tau^*$ then set $\{\Theta^* \leftarrow \Theta_{sm}^*, \tau^* \leftarrow \tau_{sm}^*\}$. Otherwise, if $\tau_{split}^* < \tau^*$ then set $\{\Theta^* \leftarrow \Theta_{split}^*, \tau^* \leftarrow \tau_{split}^*\}$. Otherwise Θ^* and τ^* are assigned as the output.
8. Threshold the rate-of-convergence of the total absolute error defined in Eq. (3.34). If the constraint is not met then go to step 3. Otherwise, halt the process and output Θ^* as the final result (the threshold is set to 0.2 in our experiment).

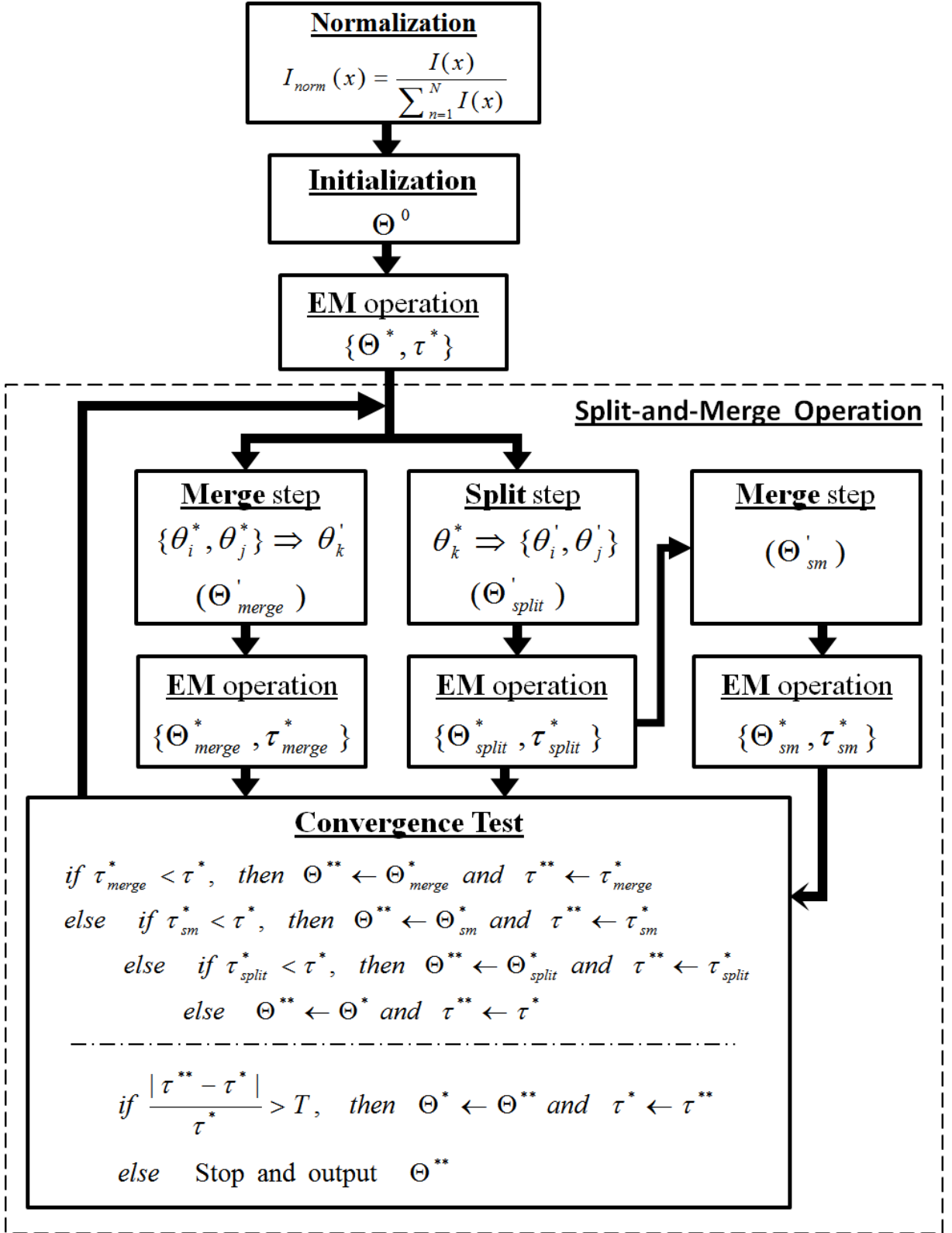


Figure 3.13: Outline of eSMEM shape modelling process

3.6 Conclusion

In this chapter, a novel application of GMMs for fitting the intensity profile of spots in a given image in order to detect protein particles has been presented. The goal of the work is to measure the colocalization of two different types of protein particles in a given image, where the proteins of each type are imaged in gray-scale and stored at different channel of the given image. However, some of the particles are too close to each other and become the cluster spots in the image which have ‘merged’ appearance in the intensity profile. This makes the colocalization analysis complicated as the candidate particles in the ‘merged’ profile are difficult to be determined.

The new GMMs approach not only estimates the number of candidate particles in a cluster spot, but also parameterize the shape characteristics of the particles for the later co-localization analysis. Since the optimization process for the parameters of the GMMs is intractable, the SMEM algorithm is applied to estimate the parameters of the mixture components of the model as well as the number of components. In order to adapt the SMEM algorithm for shape modelling and withdraw the randomness of the existing SMEM techniques, a new eSMEM algorithm that is based on the fitting error has been introduced for optimizing the parameters of the GMMs. Unlike the conventional SMEM algorithm for data point clustering, the eSMEM is specialized for shape modelling and is deterministic, i.e. repeatable. The novelty of this algorithm is the inherent EM estimation process and the split-and-merge operation that are closely related to the shape information of the given target spot. The introduction of a new measure, the error distribution, enables a new deterministic solution to the under-determined problem of component splitting, and leads to the new split and merge criteria for shape modelling.

This new approach is the basis of a practical system for spot modelling in this work, which is called Spotnick. Since the eSMEM algorithm is developed for ideal conditions, in order to model the real image data effectively, pre-processing is necessary to removed the ‘impurity’ from the acquired image data that arise during the imaging process. The pre-processing along with the modelling results for Spotnick are presented in the next chapter. This includes the co-localization analysis that measures the estimated parameters of the protein particles after shape modelling.

4

Spotnick: a practical eSMEM shape modelling tool¹

The new eSMEM algorithm uses Gaussian mixture models (GMMs) to fit and hence parameterize the intensity profile of the given spot objects. Different from the existing SMEM algorithms, the eSMEM algorithm uses the estimation error induced from the ‘target’ shape to regulate the EM process and to develop the new deterministic error-based split-and-merge algorithm. The theoretical detail of the eSMEM algorithm has been described in the previous chapter. However, in practice, the real image data of the protein particles are not ideal for the shape modelling algorithm. In order to achieve good shape modelling performance, a pre-processing stage is required to modify the image data before the modelling process.

Ideally, the eSMEM modelling framework can apply to a given image by considering the image as one ‘big’ GMM. However, this approach is impractical since the size of the image is large which requires huge computation for the modelling process. Instead, the possible spot objects are detected from the given image and the detected separate spot objects are divided from the image, and each spot patch is fitted with a GMM.

Also, the original images obtained from the confocal microscope have two problems that reduce the performance of the eSMEM shape modelling framework. The first problem is, shown in Fig. 4.1(a). Stripe-like noise existing in the intensity profile of the spots which is induced by the scanning direction of the microscope. Since the GMMs have smooth profiles, this noise can reduce the accuracy of the modelling result when fitting the GMMs to the spots in the original images.

The second problem is the irregular luminance over images, i.e. the original images contain

¹This work is presented in part in [46, 77, 78]

bright regions and dim regions. This causes difficulty for the spot detection because the possible spots can not be distinguished from the noise directly based on the intensity values. The uneven luminance is induced by the ‘background’ fluorescent markers. As mentioned in the previous chapter, an observed image is a 2D projection of the test sample with 3D structure. The fluorescent markers at lower levels than the intended scanning level appear background brightness in the observed image, and the regions with more fluorescent markers at lower levels result in higher background luminance in the image. In this case, the intensity profile of a spot in the bright region is not the actual shape profile of the corresponding particle but is biased by the ‘background’ particles. This means simply modelling the intensity profile of the spot extracted from the original image may not match the ideal parametrization of the corresponding protein particles, especially for the spots in the bright regions.

In order to detect the spots with irregular luminance, and to modify the intensity profile of the detected spots so that the shape of the spot is closer to the actual shape of the associated protein particle, a pre-processing step is applied to the given image before the eSMEM modelling process. This step consists of the noise reduction, spot detection and background subtraction. The techniques adopted in these steps are firstly described in *Section 4.1*. The complete shape modelling tool (both the pre-processing and eSMEM modelling) is tested with some synthetic ground-truth Gaussian mixture samples. In *Section 4.2*, the test results are presented and compared to the results of the state of art SMEM algorithm proposed by Zhang *et al.* [99] on the synthetic ground-truth samples. The results of the eSMEM algorithm on real data are presented and discussed in *Section 4.3*. Finally, the parameterized positions and sizes of the protein particles based on the GMMs fitted to the particle objects of different types of proteins are analyzed for colocalization which is described in *Section 4.4*.

4.1 Pre-processing Stage

Recall that the eSMEM shape modelling algorithm described in *Chapter 3* assumes the input spot represents the actual shape of the corresponding protein particle or particle cluster that is approximately a GMM with reasonable number of mixture components. In fact, each of the observed images contains a large amount of spots, as well as the noise induced during the scanning process, and is affected by the ‘background’ luminance caused by the fluorescent markers at the ‘sub-observation’ levels. The purpose of the pre-processing stage is to remove the ‘defects’ inherent in the real data so that the input data for shape modelling is closer to the ideal condition and hence achieves effective and accurate modelling results. Consequently, the task in this stage is to reduce the scanning noise, divide the separated spots in the original image into individual spot patches, and remove the ‘background’ luminance from each spot patch before shape modelling. Noise reduction is the first step of the pre-processing and is described in the following section.

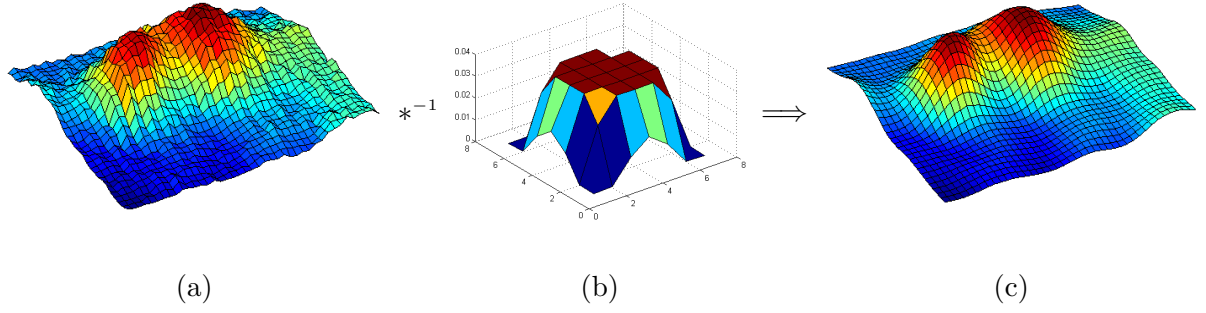


Figure 4.1: De-blurred Image Sample (a) the profile of a sample region of raw image; (b) the profile of a 7x7 circular averaging filter used as the ‘approximated’ PSF; (c) the de-noised and de-blurred profile of (a) by de-convolving the raw image with (b).

4.1.1 Noise Reduction

The first pre-processing step is to reduce the scanning noise (Fig. 4.1(a)) by deconvolving the original image I_o is with an ‘approximate’ point spread function (PSF) F_{PSF} using the Wiener deconvolution.

$$I = I_o \{ *^{-1} \} F_{PSF}$$

The reason of using deconvolution instead of convolving the image with denoising filters is that the confocal microscopy image is the result of the convolution of the ‘lateral image’ of particles by the point-spread-function of the microscope during the scanning [41, 97]. In this step, the ‘approximate’ PSF (as shown in Fig. 4.1(b)) is chosen as a circular averaging filter which gave a good image appearance (as shown in Fig. 4.1(c)) after deconvolution. The use of this PSF is clearly only an approximation since it does not model the aberration of the image. The goal of this pre-processing step is to reduce the scanning noise, and is achieved by de-noising procedure inherent in the Wiener deconvolution. Also, since a PSF with small size is applied in the deconvolution, the affect on shape of the spots is minor. Hence, the resultant spots are smoothened without much change to the shape, and is sufficient for shape modelling. The deconvolution is accomplished by the Wiener deconvolution function ‘`deconvwnr`’ in Matlab. The deconvolved image shown in Fig. 4.1(c) has a smooth profile when compared to the rough appearance in the raw image in Fig. 4.1(a). However, the intensity of each pixel in the image is changed after the deconvolution, and possibly outside the range of $[0, 255]$. Hence, the intensity of the denoised image is scaled and shifted to fit the correct range of the original image.

4.1.2 Spot Segmentation

This step segments the separate spot clusters from the image in order to reduce the computational load of the model fitting. As shown in Fig. 4.1, the ‘spot clusters’ where the spots are concentrated are distinguishable from each other in the image, and hence it is reasonable to isolate each spot-region and model the profile as a mixture of Gaussian distributions separately. However, due to uneven background luminance, the spot-cluster segmentation cannot be

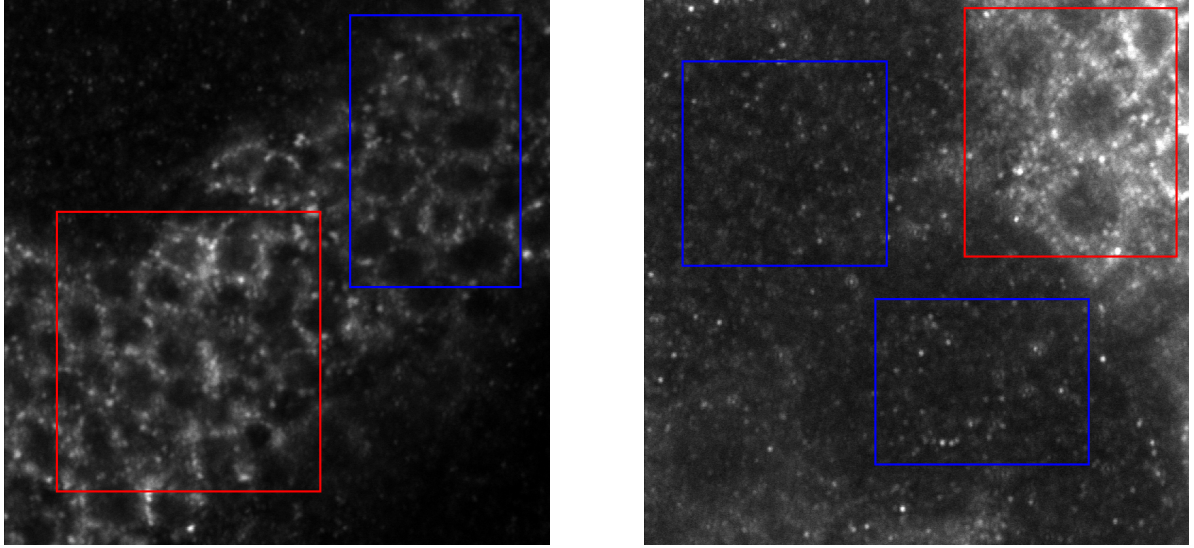


Figure 4.2: Samples of inter-regional background brightness problem (*The left image has slight brighter background in the red region while comparing to the blue region; the right image has more significant differences between the red and blue regions.*)

simply achieved by intensity thresholding of the image. Instead, the spots are detected by the contrast of the objects. Hence, the image is transformed to a contrast domain, and a contrast enhancement is performed to reduce the difference between the spots with different background luminance. The locations of the spots can then be detected by thresholding the contrast enhanced image. Each rough segmentation is expanded to cover the entire spot area and then extracted from the image for modelling.

4.1.2.1 Contrast Enhancement

As shown in Fig. 4.2, the given observed images contain regions that have different background brightness. The left image has slightly brighter background in the red region in comparison with the blue region, and the differences are more significant in the right image. The irregular background is induced by the ‘background fluorescent markers’. The observed specimen is in fact a 3D object, but the 2D image is only captured at a particular depth of the specimen. The fluorescent markers that are beyond the observation level would also emit light during the scanning, and result in background luminance in the scanned image. This would cause problems when thresholding the intensity for particle spot segmentation, where this approach favours the brighter regions over the dimmer regions. When selecting the spots manually by biologists, the spots in different regions are detected depending on the ‘sharpness’, i.e. the local contrast, which is invariant to the background luminance. In the view of signal processing, this ‘sharpness’ can be considered as the Teager-Kaiser signal energy [53, 89] of the signal. The Teager-Kaiser

energy y for a 1D signal x at the n^{th} interval is as follow

$$y(n) = x^2(n) - x(n-1)x(n+1) \quad (4.1)$$

Based on the 1D energy operator, several methods [13, 67] have extended this approach to extract the energy of 2D signal, by applying the operator in Eq. (4.1) along both the vertical and horizontal directions in the 2D domain, and the resulting signal energy can also be considered as the local contrast of the image information.

In this work, the local contrast ψ is obtained by a modified 2D energy operator, which applies Eq. (4.1) to the vertical, horizontal and diagonal directions of the image as follows

$$\begin{aligned} \psi(I(h, k)) = 4I^2(h, k) & - I(h-1, k-1)I(h+1, k+1) \\ & - I(h-1, k)I(h+1, k) \\ & - I(h-1, k+1)I(h+1, k-1) \\ & - I(h, k-1)I(h, k+1) \end{aligned} \quad (4.2)$$

The second step is to modify the contrast value nonlinearly using a gamma operation which is described by

$$\psi_a(I(h, k)) = \text{sign}(\psi(I(h, k)))|\psi(I(h, k))|^a \quad a \in (0, 1) \quad (4.3)$$

This operation reduces the contrast values non-linearly, where the higher values are reduced less while the lower values are reduced much more. Hence, the differences of the contrast between the particle spots become narrower when the differences between the spots and the noise are still large. An example of the output is shown in Fig. 4.3(c).

The reason for performing the gamma operation on the contrast values instead of the original data is as illustrated in Fig. 4.4. Performing the non-linear modification directly on the image intensity can satisfy the purpose of reducing the differences between the spots, however this approach can not solve the background luminance problem, but instead bring up the brighter background (Fig. 4.4(c)). Hence, the later thresholding has failed to detect the correct positions of the spot regions. On the contrary, this would not happen when the non-linear modification is performed on the image contrast which can reduce the spot differences and at the same time emphasize the boundaries of the spots.

The last step in defining the spot-regions is to threshold the modified-contrast image with a user defined threshold. This allows the user to have a degree of freedom to select the spot objects of interest. The thresholded image (in Fig. 4.5) is called a contrast mask and indicates the positions of the spots and a rough region mask of the spot-regions. Based on the ‘contrast mask’, the process of extracting the complete spot-regions is described in the next section.

4.1.2.2 Region Expansion and Labelling

In order to model the separated spots individually, the spots are firstly segmented from the image and assigned a unique label for each spot. From the previous step, the contrast mask

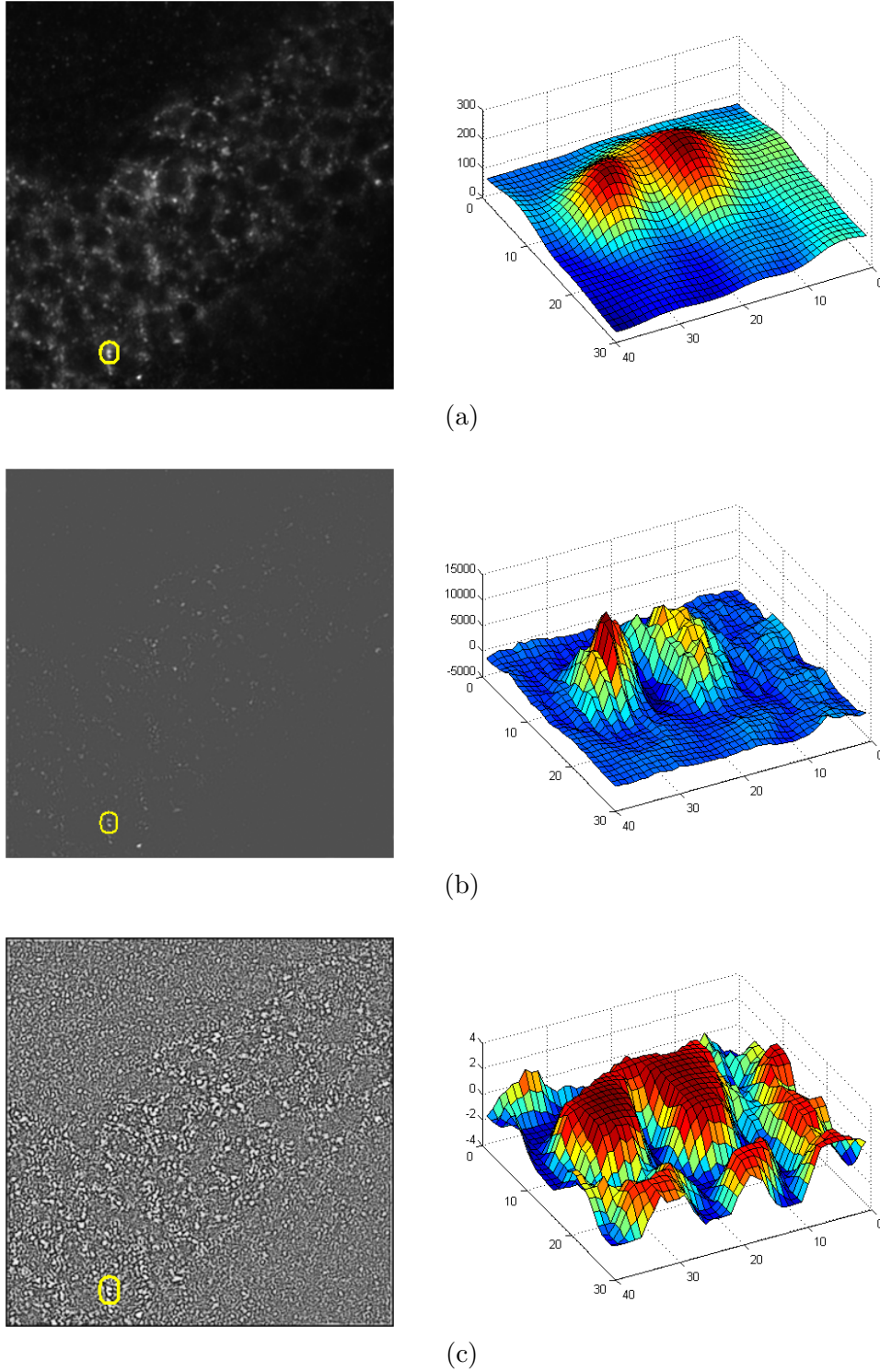


Figure 4.3: Non-linear contrast modification of the De-blurred Image. (a) *de-noised and de-blurred image and the profile of a sample spot region*; (b) *contrast image obtained from (a) using Eq. (4.2)*; (c) *non-linear modification of (b) using Eq. (4.3)*. (The right column are the profiles of the yellow regions of the images in the left column.)

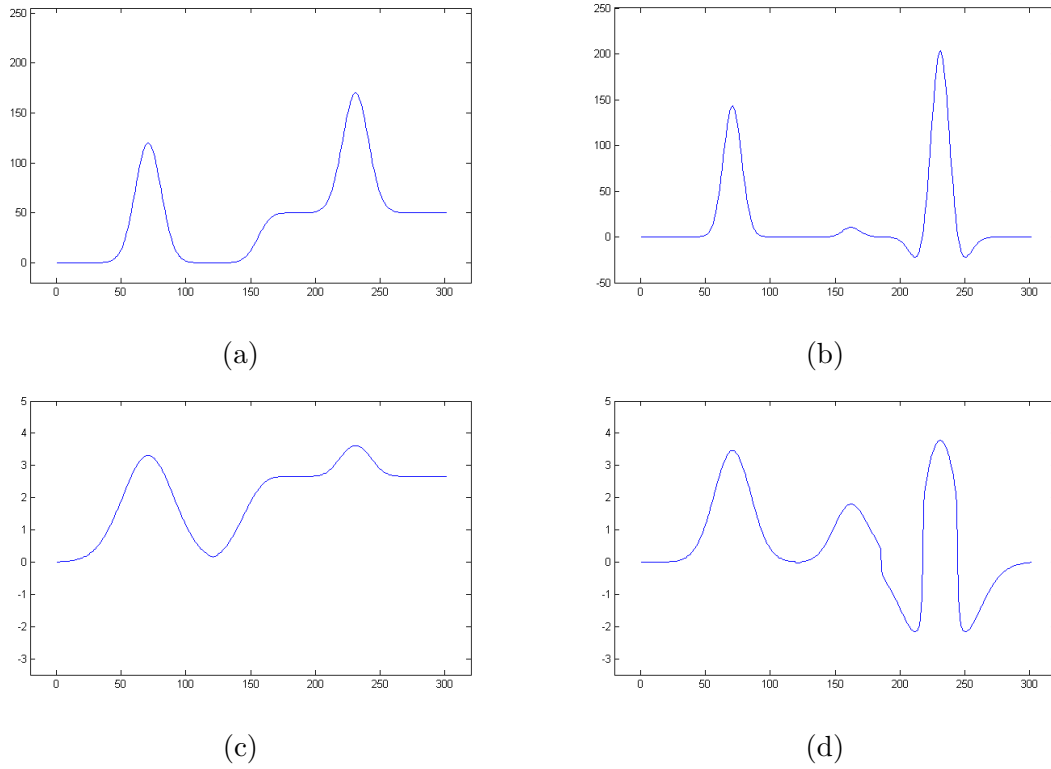


Figure 4.4: Contrast Modification Example. (a) example of two identical normal distributions with different background amplitudes; (b) contrast $\psi(\cdot)$ of (a) using Eq. (4.3); (c) direct non-linear modification on the intensity (I) of (a); (d) non-linear modification on the contrast value $\psi(\cdot)$. (Note that the direct non-linear modification on the original data also brings up the background, but it is not the case when modifying the contrast value instead.)

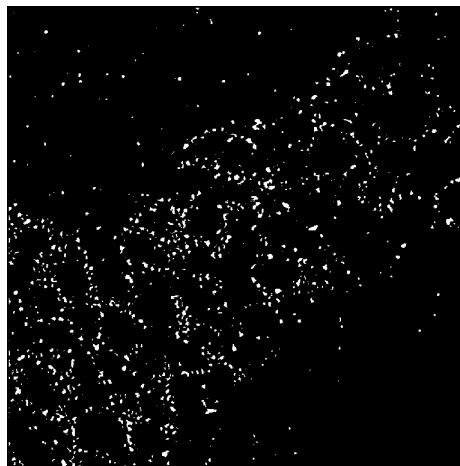


Figure 4.5: Contrast Mask. (The contrast mask is obtained by thresholding the non-linearly modified contrast value with a user selected threshold.)

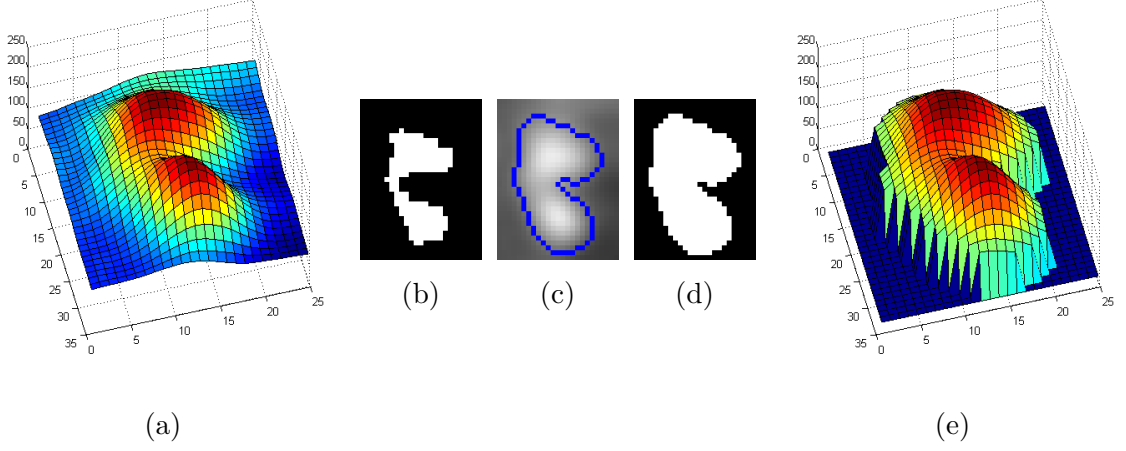


Figure 4.6: Region-Expansion Sample. (a) profile of a sample region; (b) the initial spot-area of the sample region extracted from the contrast mask; (c) the zero-crossing boundary of the sample region defined by convolving the image with a 2D DoG filter; (d) expanded spot-region area from (a) and bounded by the boundary defined in (c); (e) profile of the sample region filtered by the expanded region mask in (d).

is an incomplete region mask as it tends to under-determine the shape of each cluster. Hence, an additional step is performed to dilate the contrast mask around the segmented spot-clusters so that it can cover sufficient content of each spot-cluster for shape modelling. The edge of a spot-cluster for dilation is simply defined by the Laplacian ($\mathbf{L}$) [17] of the de-noised image $\mathbf{I}$.

$$L(h, k) = \frac{\partial^2 I}{\partial h^2} + \frac{\partial^2 I}{\partial k^2}$$

In this work, a 5x5 Difference of Gaussian (DoG) filter [64] with $\sigma_1 = 1.5$ and $\sigma_2 = 1.6$ is used to approximate the Laplacian of Gaussian (LoG) operation which is computationally efficient, and the resulting spot-patches provide enough data for modelling.

For each initial spot-area in the contrast mask, a rectangular block of the original image corresponding to the spot is extracted. The size of the block is three times the minimum rectangle containing the spot-area and acts as a limit on the size of the dilated spot region. Each spot-area expands until it reaches the zero-crossing boundary of the DoG in the corresponding image block. A sample of region expansion is shown in Fig. 4.6.

In the case of ‘broken’ boundaries, the dilation process is constrained by the expansion limit. However, some of the spots that are separated in the contrast mask may overlap with other spots after the dilation, which means the spots are very close to each other and should be considered as one cluster spot. Hence, the overlapped areas are merged and become one spot patch. Finally, the expanded spot patches are, as shown in Fig. 4.7, labelled with unique numbers using the labelling technique introduced by Haralick and Shapiro in [43].

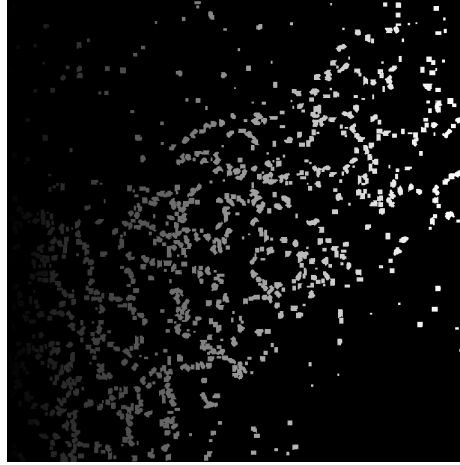


Figure 4.7: Labeled Region-extraction Mask (*The overlapped expanded regions are merged and become a single region. Each resulting region is labelled by its own number for spot-region separation.*)

4.1.3 Local Background Subtraction

After the region extraction process, the spot-regions have been defined and isolated from the image, but it requires one more step in the pre-process stage before shape modelling, which is the local background subtraction. Again, since the background luminance is affected by the ‘background fluorescent markers’, the intensity values of the spot-cluster is not just the actual brightness of the corresponding particles but also the ‘background fluorescent markers’.

This means, after extracting a spot-region by filtering the image with the corresponding region extraction mask, the shape of the spot-region would be isolated but retains the local background information as a ‘base’ (Fig. 4.8) of the profile. While the GMM modelling considers the profile of the input spot patch to be the true shape of the associated protein particle or particle-cluster, the ‘base’ reduces the modelling precision due to the intensity discontinuity at the spot boundary with the background profile. In order to retain the accuracy of the shape modelling, the ‘base’ is removed from the extracted spot patch before the modelling process. When looking at the images with different inter-regional background brightness in Fig. 4.2, the background intensity is distributed smoothly over the image when compared to the intensity profile of an isolated spot patch. Thus, the local background $\mathbf{B}$ of each extracted spot patch is sensible to be considered as a flat plane, and be estimated by using a 2D first-order polynomial as follows

$$B(h, k) = b_1h + b_2k + b_3 \quad (4.4)$$

where $\{b_1, b_2, b_3\}$ are the polynomial coefficients. These coefficients are estimated in each image patch using the least square error (LSE) method. The profiles of a sample of extracted spot patch, the estimated ‘base’ of the spot patch as well as the local background subtracted spot patch are shown in Fig. 4.8. The local background subtracted spot patch is also called the

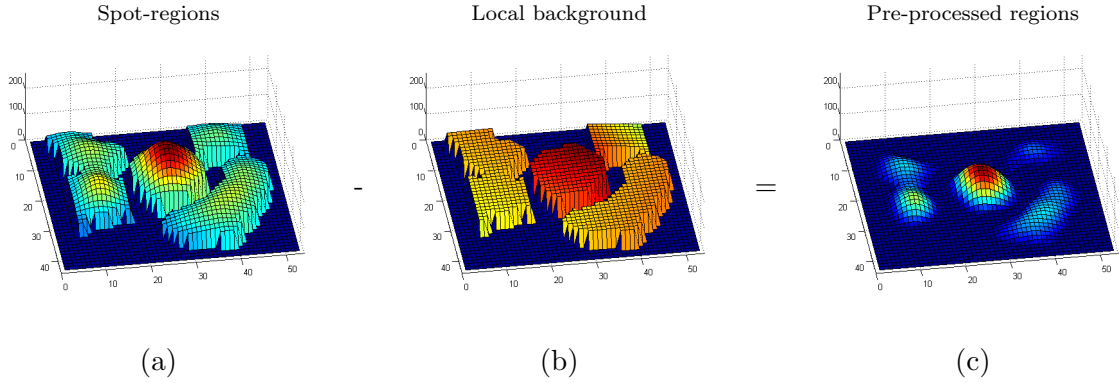


Figure 4.8: Background Subtraction. (a) profile of a sample of spots resulted by filtering the de-noised image with the region-extraction mask (b) the estimated local backgrounds using the polynomial equation in Eq. (4.4); (c) the background subtracted spots obtained by subtracting (b) from (a) which is also referred to the pre-processed spot patch.

pre-processed spot patch and is the output of the pre-processing stage. Moreover, the partial spots at the border of the image may cause failure to the background subtraction. However, such spots are removed from the modelling process as they do not have complete shapes in the image and hence are not of interest to the biologists. Also, the effect of such partial spots to the analysis is negligible as their number is much less than the complete spots in the image.

Recall that the work described so far in this chapter is the pre-processing stage for shape modelling, and in summary, the work flow of the pre-process is shown in Fig. 4.9. The output of the pre-process stage, as shown in Fig. 4.8(c), is subjected as the input to the eSMEM shape modelling process that has been described in *Chapter 3*. The complete shape modelling tool, both the pre-processing stage and the eSMEM modelling, is named ‘Spotnick’, and the results of testing Spotnick with the synthetic Gaussian mixture samples as well as the real data are presented in the following sections. Also, the modelling results of the new eSMEM algorithm is compared to the results of the Zhang *et al.* SMEM algorithm [99] which has been reviewed in *Chapter 3*.

4.2 Shape Modelling Results

In order to examine the validity and performance of Spotnick for spot shape modelling, the algorithm is tested on several synthetic ground-truth samples. Then, the performance of the new eSMEM algorithm is compared to the implementation of Zhang *et al.* SMEM algorithm [99], where the Zhang *et al.* algorithm is applied to the EM shape modelling algorithm described in *Section 3.5.1* in *Chapter 3*. The Zhang *et al.* SMEM algorithm is chosen for the comparison as it has been adopted in several recent modelling applications for data clustering [60, 63]. The eSMEM and the Zhang *et al.* SMEM algorithm are tested with a large amount of synthetic

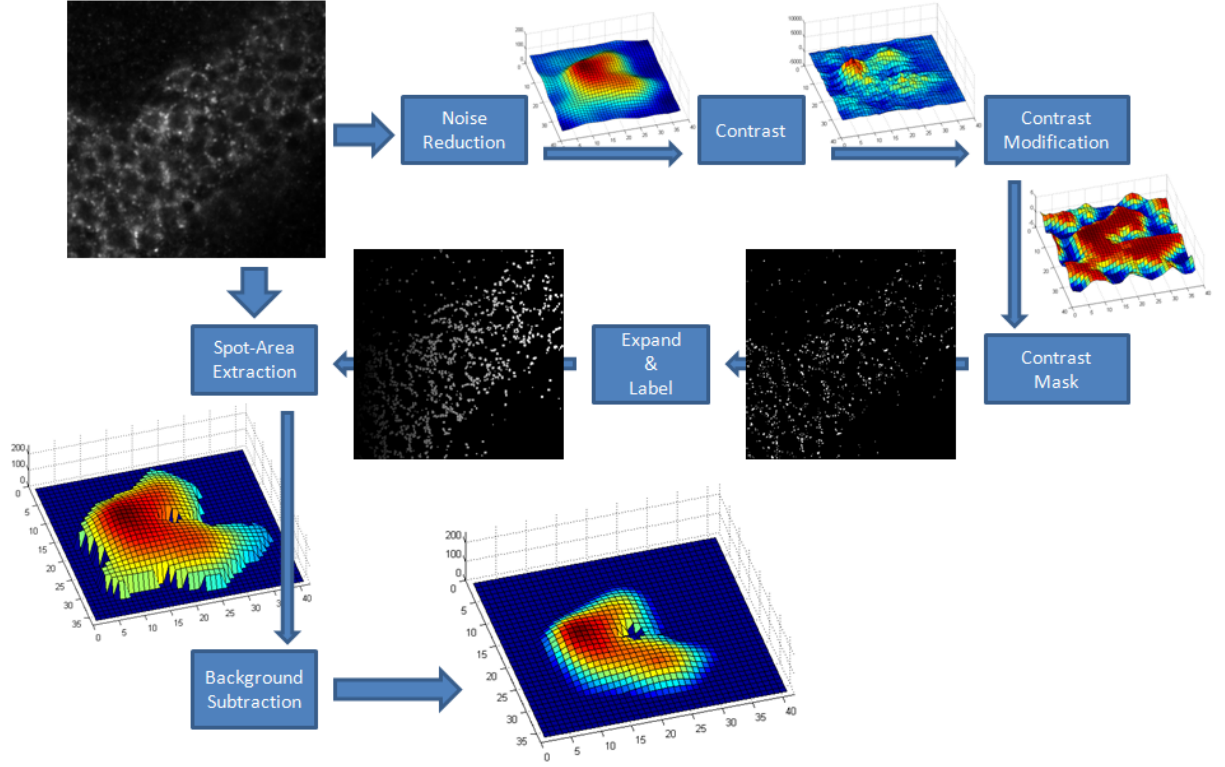


Figure 4.9: Work flow for pre-process stage (*The input is a raw gray image of one of the channel of a given double-stained image, and the outputs are the isolated pre-processed spot-regions.*)

GMMs with different numbers of mixture components.

4.2.1 Ground-truth Evaluation on Synthetic Samples

The performance of Spotnick is firstly examined by testing it on two sets of synthesized Gaussian mixture models with non-overlapping components (i.e. the initially estimated order will be the same as the true order, $M^{(0)} = M$) and the models with overlapping components (i.e. the initially estimated order will be less than the true order, $M^{(0)} < M$) respectively. The models in each set have different numbers of mixture components, which ranges from 2 to 8, i.e. $M \in [2, 8]$. Hence each set of the test models consists of 7 sub-sets, where each sub-set has 100 randomly sampled Gaussian mixture models with same number of M . Also, since the real data has background noise, Spotnick is also tested with noisy mixture models to examine the robustness to noise. The noisy models are generated by adding white Gaussian noise (i.e. background noise) to the images of the synthesized models. Since the average peak-signal-to-noise ratio (PSNR) of the real images is 31dB (which is estimated from the flat regions in the images), the PSNRs for the noisy models are set to 31dB. Consequently, Spotnick is tested by four sets of synthesized models in different scenarios, which are the non-overlapping and overlapping scenarios and the noisy versions of the two scenarios.

Evaluation metrics: The goal is to parameterize the protein particles in the image for co-localization analysis, hence the evaluation of Spotnick in this task examines the accuracy of the shape fitting, the detection of the correct model order, position accuracy and size/shape accuracy of the estimated mixture components from the modelling results. Also, since the positions are indicated by the means of the components, and the sizes/shapes rely on the mixing weights and covariance of the components, the evaluation of the modelling results is based on the following metrics. ‘ $\mathbf{M}$ ’ indicates the number of the subset which is also the number of components of each synthesized model in that subset. ‘ τ_{ave} ’ is the average value of the error-signal-ratio τ (as described in Eq. (3.25)) over the 100 testing models and is listed as a percentage. This shows the accuracy of shape fitting of the modelling algorithm for the different groups of testing models, and the lower value of τ_{ave} means the better performance in shape fitting for the modelling algorithm. The metric that evaluates the performance of model order detection in each mixture model is ‘ $N_{M'=M}$ ’. $N_{M'=M}$ is the number of correct order detections of the modelling for each subset where the estimated number M' of components is the same as the actual number M . This is an important metric for spot analysis as one of the goals is to find the number of particles in each cluster spot, and the higher value of $N_{M'=M}$ means the higher accuracy for the algorithm in detecting the correct number of particle candidates in the cluster spots in the real data. In contrast, the nature of the inaccuracy in model order estimation is evaluated by ‘ $N_{M'<M}$ ’ and ‘ $N_{M'>M}$ ’. $N_{M'<M}$ is the number of under-fit models (i.e. the estimated model order $M' < \mathbf{M}$, or a ‘false negative’ for spot detection) over the 100 trials in each subset, and the average number of the differences between M' and $\mathbf{M}$ over all the under-fit cases is indicated by $N_{f-,ave}$. In contrast, ‘ $N_{M'>M}$ ’ is the number of over-fit models, also a ‘false positive’ for spot detection (i.e. the estimated model order $M' > \mathbf{M}$) over the 100 trials, and the average difference between M' and $\mathbf{M}$ over all the over-fit cases is indicated by $N_{f+,ave}$. The two metrics, $N_{f-,ave}$ and $N_{f+,ave}$, can show how close the modelling algorithms reach the correct detection when an incorrect number of spots is detected, and also show by how much the modelling algorithms tend to under-fit or over-fit the target models.

The evaluation of the estimated parameters is to match the estimated components to the corresponding ground-truth components. Hence, the parameter evaluation is based only on the results where the model order has been detected correctly. The first metric of the parameter evaluation is ‘ $\frac{E_\alpha}{\alpha}$ ’ on the estimated component mixing weights α' . $\frac{E_\alpha}{\alpha}$ is the average error-signal-ratio for α' , where E_α is the absolute difference between α' and the ground-truth mixing weights α . Since the sizes of components in a GMM are dominated by the mixing weight, $\frac{E_\alpha}{\alpha}$ shows the accuracy of the shape modelling in characterizing the sizes of the components of the target mixture model, which is also the accuracy in estimating the sizes/brightness of the protein particles in spot analysis. The second parameter is ‘ E_μ ’, the average offset distance of the estimated means μ' from the actual means μ . This evaluates the accuracy of the modelling algorithms in detecting the correct positions of the candidate components in a given mixture model, and the lower value is equivalent to the higher accuracy. This metric is also important

as it relates to the accuracy of measuring the positions of the protein particles in a each cluster spot in real data. The shape of each mixture component is represented by the covariance Σ of the Gaussian component, and the accuracy of the estimated covariance Σ' is evaluated by ' $\frac{E_\sigma}{\sigma}$ ' and ' E_ρ '. Since each covariance $\Sigma = \begin{bmatrix} \sigma_h^2 & \rho\sigma_h\sigma_k \\ \rho\sigma_h\sigma_k & \sigma_k^2 \end{bmatrix}$ is a matrix, where σ_h and σ_k are the standard deviations along the vertical and horizontal axis respectively, and ρ is the correlation factor of Σ , the evaluation of Σ' is based on the average standard deviation $\sigma = \frac{\sigma_h + \sigma_k}{2}$ and ρ separately. Hence, $\frac{E_\sigma}{\sigma}$ is the average of absolute error-signal-ratio for the estimated σ' when compared to the actual σ , and E_ρ is the average absolute difference between the estimated ρ' and the ground-truth ρ . The lower the values of these metrics mean the closer that the shape of the estimated component is matched to the ground-truth, and the higher accuracy in characterizing the shape information of the protein particles. For the testing models with Gaussian noise, each synthesized model is firstly smoothened by using the noise-reduction step as described in *Section 4.1.1*, and the estimated results of the smoothened models are evaluated by matching to the clean versions of the corresponding test models.

The evaluations of the modelling results on the models in different scenarios are shown in the top halves of Table 4.1, Table 4.2, Table 4.3 and Table 4.4 respectively. Also, an example of the non-overlapping model and a sample of the overlapping models are shown in Fig. 4.10 and Fig. 4.11 respectively. The results for the clean models (without Gaussian noise) show that the estimations of the eSMEM algorithm for the non-overlapping models are very close to the ground-truth where the fitting errors (e.g. τ_{ave}) are very low for the different models orders, and the average errors in the estimated parameters are negligibly small. However, both the fitting errors and the parameter errors become obvious in the results for overlapping models (see Table 4.3). This is because when the components are non-overlapped, the information of each component in the mixture is sufficient for the algorithm to estimate the corresponding parameters, and more importantly the initialization can detect the true positions (i.e. the means) of the non-overlapped components so that the modelling algorithm can converge the rest of the parameters to the global optimum much easier. For the results of noisy models in Table 4.2 and Table 4.4, the error of the estimated shape and parameters have high values when compared to Table 4.1 and Table 4.3 respectively are due to the distortion of the synthesized model shapes caused by the noise. The noise reduces the precision of Spotnick in the detecting the correct order of the testing models, where some noise can trigger false positives in the component detection (see $N_{f+,ave}$), and in contrast some noise located between close components can lead to false negative detections (see $N_{f-,ave}$) by detecting the close components as one single component.

Comparison to the initial parameters: In order to examine the improvement of the estimation over the initial parameters estimated by using the methods described in *Section 3.5.2* in *Chapter 3*, the evaluation on the initial parameters of the non-overlapping models and the noise versions are also shown in the bottom halves of Table 4.1 and Table 4.2 respectively, which are compared to the results of the eSMEM algorithm. Also, the initialization and the eSMEM modelling result of a clean synthesized model with $M = 8$ are shown in Fig. 4.10. As shown in

	$\mathbf{M}$	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_{\alpha}}{\alpha}$ (%)	E_{μ}	$\frac{E_{\sigma}}{\sigma}$ (%)	E_{ρ}	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
eSMEM	2	0.003	100	4.4×10^{-4}	0	2×10^{-3}	9.1×10^{-6}	0	0	0	0
	3	0.004	100	7.4×10^{-4}	0	2.3×10^{-3}	8.2×10^{-6}	0	0	0	0
	4	0.007	100	1.4×10^{-3}	0	4.1×10^{-3}	1.8×10^{-5}	0	0	0	0
	5	0.007	100	1.6×10^{-3}	0	4.1×10^{-3}	1.6×10^{-5}	0	0	0	0
	6	0.008	100	1.9×10^{-3}	0	4.8×10^{-3}	2.2×10^{-5}	0	0	0	0
	7	0.01	100	2.3×10^{-3}	0	5.7×10^{-3}	2.4×10^{-5}	0	0	0	0
	8	0.011	100	2.4×10^{-3}	0	6×10^{-3}	2.5×10^{-5}	0	0	0	0
	8	0.011	100	2.4×10^{-3}	0	6×10^{-3}	2.5×10^{-5}	0	0	0	0
Initialization Only	2	28.16	100	8.67	0	16.76	0.24	0	0	0	0
	3	33.13	100	10.33	0	19.43	0.25	0	0	0	0
	4	37.05	100	10.40	0	21.93	0.26	0	0	0	0
	5	39.45	100	11.70	0	23.36	0.25	0	0	0	0
	6	40.87	100	11.01	0	24.52	0.24	0	0	0	0
	7	44.10	100	11.64	0	26.41	0.25	0	0	0	0
	8	46.60	100	11.99	0	27.82	0.25	0	0	0	0
	8	46.60	100	11.99	0	27.82	0.25	0	0	0	0

Table 4.1: Result evaluation based on synthesized models with non-overlapping components. (The top half of the table shows the evaluation on the results of the eSMEM algorithm and the bottom half shows the evaluation on the results of the initialization (i.e. $\Theta^{(0)}$) using the methods described in Section 3.5.2 in Chapter 3. The tests are based on the synthetic Gaussian mixtures models of order $\mathbf{M}$ but the components are non-overlapping in each model, i.e. $M^0 = \mathbf{M}$. The evaluation listed in the table is based on 100 trials of randomly sampled Gaussian mixture models. ‘ τ_{ave} ’ is the average value of the error-signal-ratio of each modelling estimate. ‘ $N_{M'=M}$ ’ is the number of correct order detections (i.e. the estimated model order $M' = \mathbf{M}$) over the 100 trials. ‘ $\frac{E_{\alpha}}{\alpha}$ ’ is the average error-signal-ratio for the estimated component mixing weights α' . ‘ E_{μ} ’ is the average offset distance of the estimated means μ' when comparing to the ground-truth. ‘ $\frac{E_{\sigma}}{\sigma}$ ’ is the average absolute error-signal-ratio for the average standard deviation σ' of the estimated covariance Σ' , and ‘ E_{ρ} ’ is the average absolute difference between the estimated ρ' of Σ' and the ground-truth ρ . ‘ $N_{M'<M}$ ’ is the number of under-fit models (i.e. the estimated model order $M' = \mathbf{M}$) over the 100 trials, which can also be considered as the ‘false negative’ rate, and $N_{f-,ave}$ is the average number of the differences between M' and $\mathbf{M}$ over all the under-fit cases. ‘ $N_{M'>M}$ ’ is the number of over-fit models, also the ‘false positive’ rate (i.e. the estimated model order $M' = \mathbf{M}$) over the 100 trials, and $N_{f+,ave}$ is the average number of the differences between M' and $\mathbf{M}$ over all the over-fit cases.)

	$\mathbf{M}$	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
eSMEM	2	12.89	100	6.41	0.34	32.40	0.22	0	0	0	0
	3	15.36	94	6.71	0.47	34.98	0.25	5	1	1	1
	4	18.67	92	7.28	0.67	37.57	0.28	5	1	3	1
	5	25.77	85	9.30	0.89	39.36	0.28	13	1	2	1
	6	30.43	84	9.36	1.24	39.03	0.28	11	1	5	1.2
	7	35.34	82	11.10	1.59	38.08	0.31	14	1	4	1.6
	8	41.72	70	11.75	2.15	35.57	0.34	23	1	7	1.5
	8	41.72	70	11.75	2.15	35.57	0.34	23	1	7	1.5
Initialization Only	2	34.94	92	9.70	1.37	45.95	0.32	0	0	4	1.3
	3	39.38	83	10.51	1.92	49.02	0.33	0	0	10	1.2
	4	43.09	74	13.62	2.02	51.31	0.34	0	0	13	1.2
	5	44.62	62	14.74	2.09	52.92	0.35	0	0	18	1.4
	6	46.08	52	14.17	2.43	53.90	0.35	0	0	20	1.6
	7	49.02	45	13.68	2.72	55.64	0.35	0	0	25	1.8
	8	51.48	43	14.00	3.28	57.05	0.36	0	0	37	1.3
	8	51.48	43	14.00	3.28	57.05	0.36	0	0	37	1.3

Table 4.2: Result evaluation based on synthesized models with non-overlapping components and Gaussian background noise. (The top half of the table shows the evaluation on the results of the eSMEM algorithm and the bottom half shows the evaluation on the results of the initialization (i.e. $\Theta^{(0)}$) using the methods described in Section 3.5.2 in Chapter 3. The tests are based on the synthetic Gaussian mixtures models of order $\mathbf{M}$ but the components are non-overlapping in each model, i.e. $M^0 = \mathbf{M}$. Also, each synthetic model has Gaussian background noise in the image where the PSN ratio is 31dB. The evaluation listed in the table is based on 100 trials of randomly sampled Gaussian mixture models.).

Table 4.1, the eSMEM algorithm highly improved the shape fitting from the initial parameters $\Theta^{(0)}$ when the average estimation error τ_{ave} is decreased by an average of 99.98% from the initialization for the non-overlapping models. Also, the parameters estimated using the eSMEM shape modelling (Fig. 4.10(c)) are very close to the ground-truth (Fig. 4.10(a)), and the errors in the estimated parameters (i.e. $\frac{E_\alpha}{\alpha}$, E_μ , $\frac{E_\sigma}{\sigma}$ and E_ρ in Table 4.1) are negligible when compared to the initialization.

For the noisy models in Table 4.2, the eSMEM algorithm improves τ_{ave} by an average of 43.41%. The errors in the estimated parameters are also reduced, and more importantly, the true detection rate is increased by an average of 41.42%. This means that the eSMEM algorithm can improve the accuracy of the size/shape estimation and true order detection of the target mixture candidates from the initialization. Even though the eSMEM algorithm tends to under-fit the models, it recovers the false positive in the initialization caused by the noise.

Effect of the initialization: In order to examine the effect of the initialization stage in shape modelling and the robustness of the eSMEM algorithm over initialization, the eSMEM algorithm

	M	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
eSMEM	2	5.10	98	7.66	0.63	9.17	0.10	1	1	1	1
	3	5.01	95	9.38	0.68	7.33	0.08	4	1	0	0
	4	4.87	87	10.75	0.70	5.38	0.06	13	1	0	0
	5	5.28	92	12.07	0.81	4.95	0.05	8	1	0	0
	6	5.97	86	15.92	0.95	4.44	0.05	14	1.2	0	0
	7	7.92	75	19.72	1.24	4.54	0.04	25	1.1	0	0
	8	8.02	71	20.22	1.65	4.12	0.04	29	1.1	0	0
	8	8.02	71	20.22	1.65	4.12	0.04	29	1.1	0	0
eSMEM ($M^{(0)} = 1$)	2	5.10	98	7.66	0.63	9.17	0.10	1	1	1	1
	3	5.41	78	13.28	0.81	8.68	0.11	22	1	0	0
	4	5.56	69	16.70	1.13	8.56	0.08	31	1.1	0	0
	5	10.92	53	19.23	1.53	7.71	0.08	47	1.3	0	0
	6	17.95	45	17.83	1.48	8.64	0.09	55	2	0	0
	7	36.55	34	18.46	1.41	9.32	0.07	66	2.3	0	0
	8	29.02	29	22.16	1.66	9.64	0.08	71	3.1	0	0
	8	29.02	29	22.16	1.66	9.64	0.08	71	3.1	0	0

Table 4.3: Result evaluation based on the synthesized models with overlapping models. (The top half of the table shows the evaluation on the results of the eSMEM algorithm started with the initialization (i.e. $\Theta^{(0)}$) using the methods described in Section 3.5.2 in Chapter 3. The bottom half shows the evaluation on the results of the eSMEM algorithm started with only one component (i.e. $M^{(0)} = 1$).

is again tested with the clean and noisy overlapping models but with only one initial component (i.e. $M^{(0)} = 1$). The evaluations of the results are shown in the bottom halves of Table 4.3 and Table 4.4 respectively, and are compared to the results of the algorithm with the normal initialization stage. As shown in Table 4.3, without the normal initialization stage, the average error-signal-ratio τ_{ave} increases by an average of 136% of the result of the complete algorithm, and the true order detection rate is reduced by an average of 35%. Also, rates of under-fitting ($N_{M'<M}$) and the ‘false negative’ ($N_{f-,ave}$) of the model components increase by an average of 240% and 57% respectively of the original results. This is because the eSMEM algorithm usually can only converge the model parameters to a local optima, and the ‘distance’ between this local optima to the global optimum relies on how close the initial parameters are set to the optimal values. Similarly, for the noisy overlapping models, τ_{ave} is increased by an average of 21%, $N_{M'<M}$ and $N_{f-,ave}$ increase by an average of 60% and 43% respectively. Accordingly, the comparison in Table 4.3 shows that the initialization proposed in this task can help the modeling algorithm to achieve higher accuracy in the shape fitting and parameter estimation.

The split-and-merge process: The core of the eSMEM algorithm is the split-and-merge operation. In order to visualize the operation during the modelling process, the detail of the fitting process on the synthesized overlapping model shown in Fig. 4.11(a) is illustrated in Fig.

	M	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
eSMEM	2	27.30	84	28.93	1.28	34.46	0.36	12	1	2	1
	3	29.97	72	28.86	1.99	38.43	0.31	23	1	4	1.1
	4	33.78	65	30.71	2.41	35.69	0.29	30	1.2	5	1.1
	5	39.13	64	32.52	4.73	39.07	0.32	29	1	7	1.3
	6	40.61	51	36.91	5.65	36.01	0.33	45	1.3	4	1
	7	44.38	45	38.27	6.37	38.81	0.29	55	1.4	0	0
	8	51.83	36	40.66	8.27	38.00	0.31	64	1.5	0	0
	8	51.83	36	40.66	8.27	38.00	0.31	64	1.5	0	0
eSMEM ($M^{(0)} = 1$)	2	27.30	84	28.93	1.28	34.46	0.29	12	1	2	1
	3	37.40	64	32.03	2.53	40.83	0.33	34	1.1	2	1
	4	40.96	45	33.73	4.02	41.58	0.29	54	1.2	1	1
	5	46.49	32	39.14	4.75	42.17	0.34	66	1.4	2	1.3
	6	51.18	22	40.12	6.14	40.03	0.39	76	2	2	1
	7	59.14	10	42.31	7.20	45.09	0.41	90	2.6	0	0
	8	62.17	15	45.71	8.97	42.39	0.37	85	3.2	0	0
	8	62.17	15	45.71	8.97	42.39	0.37	85	3.2	0	0

Table 4.4: Result evaluation based on the synthesized models with overlapping models and Gaussian background noise. (The top half of the table shows the evaluation on the results of the eSMEM algorithm started with the initialization (i.e. $\Theta^{(0)}$) using the methods described in Section 3.5.2 in Chapter 3. The bottom half shows the evaluation on the results of the eSMEM algorithm started with only one component (i.e. $M^{(0)} = 1$).

4.12. As shown in Fig. 4.12, the GMM starts with one mixture component in the initialization step and the model parameters are converged through the EM process. In the 1st iteration ($i = 1$) of the split-and-merge process, the single component is split into two components at the split step and converge to a local optima through the EM process. In this case, the merge step for single mixture component is not feasible. Also, since the model always converges to the global optimum for $M = 1$ which is already estimated in the initialization step, the split-merge step for merging the two new components back into a single one is unnecessary. Hence, split-estimate is selected as it reduces the error-to-signal ratio τ , and is subjected to the next iteration.

For $i = 2$, the merge step is again unnecessary as it is not possible to get a single mixture model with a better fit. On the contrary, in the split step, the ‘blue’ component of the ‘selected’ model from $i = 1$ is appropriately chosen based on the new split-criterion and split into two so that the GMM has 3 components. After convergence, the ‘blue’ and ‘red’ components of the split estimate are chosen for merging in the split-merge step which reaches another local optima for $M = 2$. This new optima for $M = 2$ is better than the local optima achieved in $i = 1$, and because the split-merge estimate has higher priority than the split estimate in the

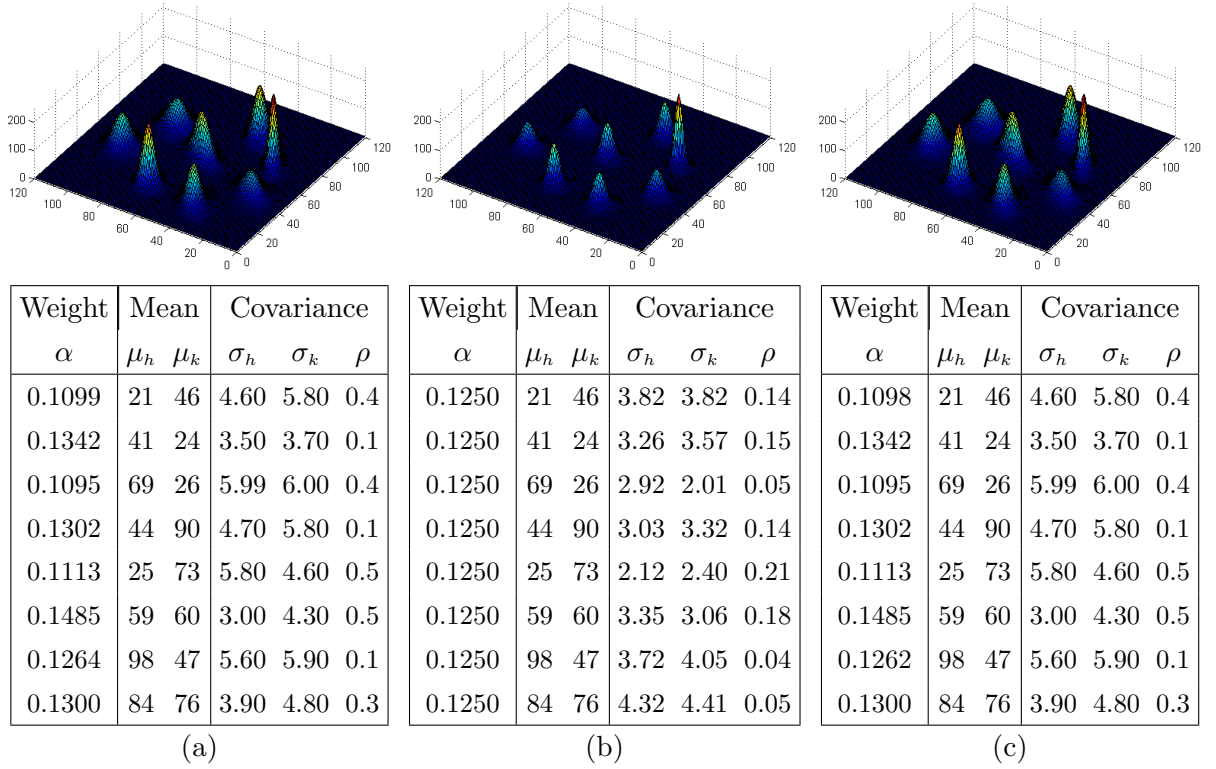


Figure 4.10: Initialization and eSMEM estimation for a synthesized Gaussian mixture model with $M = 8$ (a) a synthesized Gaussian mixture model with $M = 8$ and the corresponding parameter values. (b) the initialization of the model parameters. (c) the modelling estimations of (a) using eSMEM, and the error-signal-ratio τ is 0.02%.

prioritized split-merge selection stage as stated in *Section 3.5.7* in *Chapter 3*, the split-merge estimate ($M = 2$) is selected for the next iteration. For $i = 3$, the process is similar to $i = 2$, but this time the split-merge estimate ($M = 2$) is not chosen because it is less well fit to the shape than the input ($M = 2$) from $i = 1$. Instead, the split estimate ($M = 3$) is selected as it has better fit than the input and hence is selected as the starting point for the next iteration.

For $i = 4$, the merge step is equivalent to the split-merge step in $i = 3$ and hence is not performed. The split step split the ‘green’ component of the selected model in $i = 3$ to the new ‘green’ and ‘gray’ components. These two new components are appropriately selected for merging back into the ‘green’ component in the following split-merge step, and the converged split-merge estimate which is a better local optima for $M = 3$ is selected for the next iteration.

For $i = 5$, this time the merge step is computed by merging the ‘red’ and ‘blue’ components of the selected model in $i = 4$ into the new ‘blue’ component and reach another local optima for $M = 2$. In contrast, the split step picks the ‘green’ component of the ‘selected’ model for splitting into the new ‘green’ and ‘gray’ components and converges to another local optima for $M = 4$. In the following split-merge-step, the converged ‘green’ and ‘gray’ components are merged back again and reach a better optima for $M = 3$. The split-merge-estimate is then

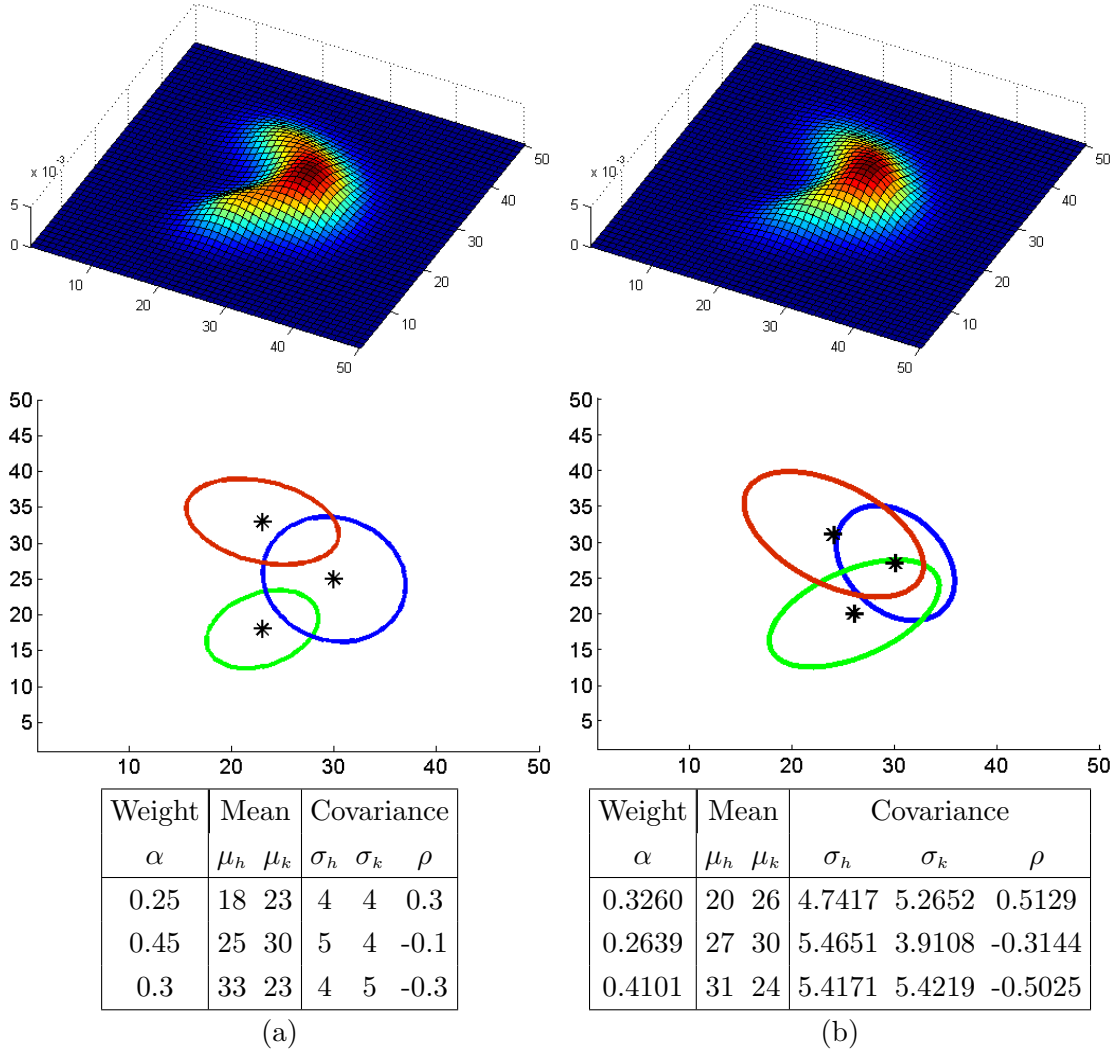


Figure 4.11: Modelling results of the eSMEM algorithm for a synthesized Gaussian mixture model with 3 overlapping components. (a) shows the groundtruth data of the synthesized model. (b) shows the estimation of (a) using the eSMEM algorithm. The top row shows the profile of the mixture models and middle row shows the corresponding contours of the components in the mixture models. Note that the black (*) are the mean of the components. The last row shows the parameters for corresponding mixture models.

selected and since the process meets the rate-of-convergence of the termination scheme that is stated in Eq. (3.34) in Chapter 3, the selected estimate is also the modelling output of Spotnick for Fig. 4.11(a). The profile of the output estimate is shown in Fig. 4.11(b). The process illustration shows that the components of the estimated model are improved at each iteration of the process, and the prioritized split-merge selection stage helps the model to converge to a better result for the same model order and hence prevents over-fitting in the modelling process.

Effect of the prioritized split-merge selection: Since a model with more components

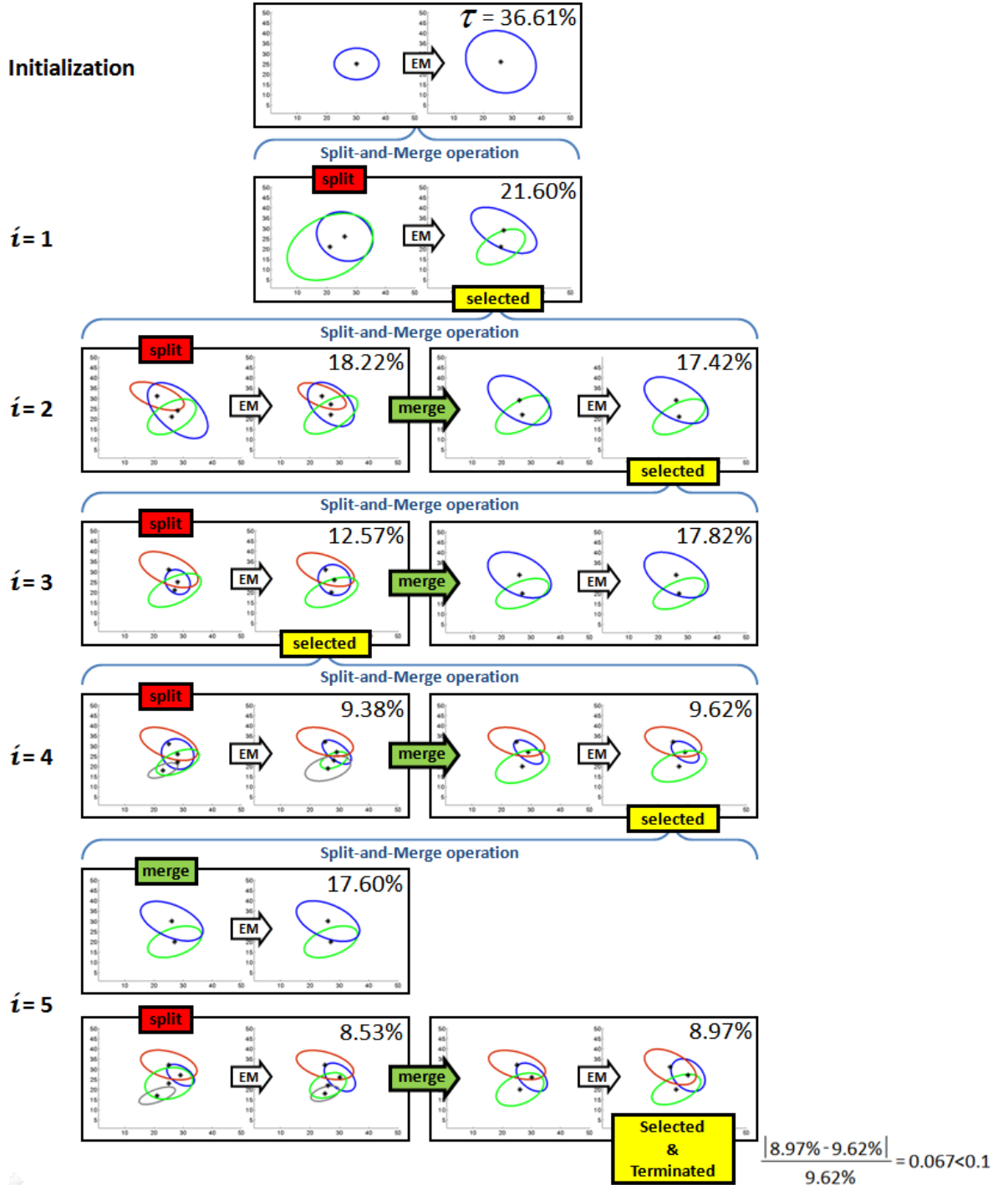


Figure 4.12: An illustration of the process of the eSMEM shape modelling on Fig. 4.11(a). This figure illustrates the operation of the iterative eSMEM process in Spotnick which is stated in Section 3.5.7 in Chapter 3. The input model of this illustration is shown in Fig. 4.11(a), and the final output of the modelling process is shown in Fig. 4.11(b). The model is shown in the contour representation for better illustration of the variation of the mixture components during the process. Also, ‘i’ indicates the iterations in the split-and-merge process.

	M	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_{\alpha}}{\alpha}$ (%)	E_{μ}	$\frac{E_{\sigma}}{\sigma}$ (%)	E_{ρ}	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
eSMEM	2	7.59	77	13.97	1.45	9.06	0.08	0	0	23	1
	3	5.97	70	10.95	1.18	7.2	0.07	3	1	27	1.5
	4	4.87	70	8.89	0.93	4.66	0.05	0	0	30	1.2
	5	5.65	59	10.23	1.21	4.42	0.07	2	1	49	1
	6	6.54	44	11.21	1.39	3.45	0.07	3	1.1	53	1
	7	6.69	40	10.4	1.31	3.61	0.09	0	0	60	1
	8	7.23	37	13.72	1.82	3.42	0.11	0	0	63	1.3

Table 4.5: Results of the eSMEM algorithm without using the prioritized selection in split-and-merge operation. *(The results are estimated from the set of overlapping synthesized models without added noise.)*

always fits the data better, the prioritized split-merge selection stage in the eSMEM algorithm can prevent the GMM having un-necessarily exceeded the number of components for fitting the target shape, and increases the chance of achieving the global optimum for the algorithm. In order to show the effect of the prioritized split-merge selection stage quantitatively, the performance of the eSMEM algorithm on the synthesized overlapping models without using the prioritized selection is shown in Table 4.5. In this case, the selection between the merge and split proposals is unbiased and is only based on the fitting error. By comparing Table 4.5 and the top half of Table 4.3, although the fitting errors (τ_{ave}) and the errors in the parameter estimation are close for the two scenarios, the one without using the prioritized selection gave less true order detection rate for higher orders, where $N_{M'=M}$ is decreased down to an average of 52%. This causes the algorithm to over-fit the models as the over-fitting rates ' $N_{M'>M}$ ' and the average 'false positive' $N_{f+,ave}$ are highly increased. This means the prioritized split-merge selection is a key to the eSMEM algorithm that prevents over-fitting and hence reduces the chance of having false positives for spot detection in the real data.

So far, Spotnick has successfully modelled the non-overlapping models and provides good estimations to the overlapping models. In order to verify the performance of the new algorithm, the eSMEM algorithm is compared to the Zhang et al. SMEM algorithm, which is the most popular of the state-of-art SMEM algorithms, and the comparison is given in the next section.

4.2.2 Comparison with the Zhang *et al.* SMEM algorithm [99]

The comparison between the eSMEM algorithm and the Zhang *et al.* SMEM algorithm examines the performance of the algorithms on the synthesized Gaussian mixture models with non-overlapping components and the models with overlapping components respectively. Similar to the previous section, the models in each scenario have different numbers (model orders) of mixture components. Also, the noise sensitivities of the two algorithms are compared by testing

the algorithms with noisy mixture models (PSNR = 31dB). Consequently, the two algorithms are tested by the four sets of synthesized models in different scenarios. Similarly, the order of the synthesized models in each set ranges from 2 to 8, i.e. $M \in [2, 8]$. Hence each set of the test models consists of 7 sub-sets, where each sub-set has 100 synthesized models with same number of M . Each model in the 1st set has non-overlapping components (i.e. $M^{(0)} = M$) and hence the initialization is close to the ground-truth parameters. This set is designed to examine the feasibility of the EM algorithm for shape modelling of the two algorithms. The models in the 2nd set also have $M^{(0)} = M$, but they have Gaussian background noise in the images of the models in order to examine the robustness of the GMM modellings to noise. The core of the comparison is to compare the performance of the split-and-merge operations of the algorithms. The performance of the different split-and-merge operations is evaluated by the 3rd testing set where each synthesized model has multiple overlapping components (i.e. $M^{(0)} < M$). Also, the noise sensitivities of the two split-and-merge operations are tested by the models in the 4th set, where each testing model has multiple overlapping components and has Gaussian background noise.

In the comparison, both algorithms use the same initialization step as described in *Section 3.5.2 in Chapter 3*. Again, for the testing models with added noise, each synthesized model is firstly smoothened by using the noise-reduction step and the estimated results of the smoothened models are evaluated by matching to the clean versions of the corresponding test models. Since the Zhang *et al.* split algorithm relies on random split proposals, the Zhang *et al.* SMEM modelling process has a selection stage in the split step where 20 splits are performed in each split step and the one with the best fit is selected as the output of that step, and is then compared to the merge result in that iteration of the split-and-merge operation. Since the evaluation of the two algorithms examines the shape fitting, the detection of the correct model order, position accuracy and size/shape accuracy of the estimated mixture component from the modelling results, the metrics used in the evaluation of the two algorithms are the same as described in the previous section. The evaluations of the two algorithms based on the 4 testing sets are listed respectively in Table 4.6, Table 4.7, Table 4.8 and Table 4.9. The comparison of the eSMEM algorithm and the Zhang *et al.* SMEM algorithm is based on considering the shape fitting accuracy, model order detection accuracy, position (the means) accuracy, and size/shape accuracy of the modelling estimations, as well as the noise sensitivity. The accuracies of the algorithm for model order detection and the estimation of positions and sizes/shapes are very important as these estimations are the keys to parameterize the protein particles for the co-localization analysis.

Shape fitting accuracy: The shape fittings of the both algorithms are very close for the non-overlapping models as the average error-signal-ratios τ_{ave} are very small and are close for the different model orders (see Table 4.6). This is because the goal of the 1st set is to test the EM algorithm for shape modelling, and the two algorithms start with the same initial parameters. The differences become larger for the models with overlapping components in 3rd testing set

	$\mathbf{M}$	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
Zhang <i>et al.</i>	2	0.002	100	2.9×10^{-4}	0	1.3×10^{-3}	3.9×10^{-6}	0	0	0	0
	3	0.005	100	9.6×10^{-4}	0	2.8×10^{-3}	9.5×10^{-6}	0	0	0	0
	4	0.007	100	1.4×10^{-3}	0	4.2×10^{-3}	2.2×10^{-5}	0	0	0	0
	5	0.009	100	2.3×10^{-3}	0	5.5×10^{-3}	2.1×10^{-5}	0	0	0	0
	6	0.008	98	1.3×10^{-3}	0	3.3×10^{-3}	1.4×10^{-5}	0	0	2	1
	7	0.01	90	2.4×10^{-3}	0	5.7×10^{-3}	2.5×10^{-5}	0	0	10	1
	8	0.014	96	3.1×10^{-3}	0	7.4×10^{-3}	2.9×10^{-5}	0	0	4	1
	8	0.014	96	3.1×10^{-3}	0	7.4×10^{-3}	2.9×10^{-5}	0	0	4	1
eSMEM	2	0.003	100	4.4×10^{-4}	0	2×10^{-3}	9.1×10^{-6}	0	0	0	0
	3	0.004	100	7.4×10^{-4}	0	2.3×10^{-3}	8.2×10^{-6}	0	0	0	0
	4	0.007	100	1.4×10^{-3}	0	4.1×10^{-3}	1.8×10^{-5}	0	0	0	0
	5	0.007	100	1.6×10^{-3}	0	4.1×10^{-3}	1.6×10^{-5}	0	0	0	0
	6	0.008	100	1.9×10^{-3}	0	4.8×10^{-3}	2.2×10^{-5}	0	0	0	0
	7	0.01	100	2.3×10^{-3}	0	5.7×10^{-3}	2.4×10^{-5}	0	0	0	0
	8	0.011	100	2.4×10^{-3}	0	6×10^{-3}	2.5×10^{-5}	0	0	0	0
	8	0.011	100	2.4×10^{-3}	0	6×10^{-3}	2.5×10^{-5}	0	0	0	0

Table 4.6: Results comparison based on the non-overlapping synthesized models. (*The top half and the bottom half of the table show the estimation results of the SMEM algorithm of Zhang et al. [99] and the new eSMEM algorithm respectively. The algorithms are tested by modelling spots created from synthetic Gaussian mixtures models of order $\mathbf{M}$ but the components are non-overlapping in each model, i.e. $M^0 = \mathbf{M}$. The evaluation listed in the table is based on 100 trials of randomly sampled Gaussian mixture models.*)

(Table 4.8), the eSMEM algorithm has better shape fitting as τ_{ave} is 15.33% on average less than the Zhang *et al.* SMEM algorithm over all the different $\mathbf{M}$. For the noisy models, although the values of τ_{ave} for the two algorithms are much higher than the estimation of the clean models, τ_{ave} of the eSMEM algorithm are still 23.2% and 17.87% less than the Zhang *et al.* SMEM algorithm on average for the non-overlapping models and overlapping models respectively. This means the eSMEM algorithm has higher accuracy in shape fitting than the Zhang *et al.* SMEM algorithm even though the performance is reduced for the noisy data.

Order detection accuracy: Since one of the goals of the shape modelling is to detect the number of candidate particles in each spot in the real image, the order detection accuracy of the modelling algorithms is an important aspect in this comparison. For the non-overlapping models, the eSMEM algorithm shows great accuracy in detecting the correct model order as $N_{M'=M}$ is 100% for all the different orders. The Zhang *et al.* SMEM algorithm also has good performance for the non-overlapping models but the accuracy decreases as the model order increases, and the algorithm starts to over-fit the models. The decrement is much larger for the overlapping models, where $N_{M'=M}$ of the Zhang *et al.* SMEM algorithm drops to 11% for $\mathbf{M} = 8$ (see Table

	M	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
Zhang et al.	2	16.15	96	7.01	1.06	38.20	0.22	0	0	4	1
	3	21.94	88	9.43	1.36	40.31	0.25	0	0	12	1
	4	25.51	74	11.15	1.59	41.10	0.28	0	0	26	1.2
	5	32.38	64	19.82	2.01	41.77	0.28	0	0	36	1.3
	6	41.13	58	22.01	2.25	40.33	0.28	2	1	40	1.9
	7	45.42	55	21.18	2.39	39.07	0.31	3	1	42	1.7
	8	50.16	38	18.90	2.94	40.31	0.34	7	1	45	1.8
eSMEM	2	12.89	100	6.41	0.34	32.40	0.22	0	0	0	0
	3	15.36	94	6.71	0.47	34.98	0.25	5	1	1	1
	4	18.67	92	7.28	0.67	37.57	0.28	5	1	3	1
	5	25.77	85	9.30	0.89	39.36	0.28	13	1	2	1
	6	30.43	84	9.36	1.24	39.03	0.28	11	1	5	1.2
	7	35.34	82	11.10	1.59	38.08	0.31	14	1	4	1.6
	8	41.72	70	11.75	2.15	35.57	0.34	23	1	7	1.5

Table 4.7: Results comparison based on the non-overlapping synthesized models with Gaussian noise. ($PSNR = 31dB$)

	M	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
Zhang et al.	2	5.53	86	23.57	2.02	5.73	0.10	6	1	8	1.4
	3	6.00	65	28.66	2.28	7.94	0.12	8	1	27	1.2
	4	7.64	41	38.26	4.69	7.64	0.15	15	1	44	1
	5	7.04	34	23.44	3.42	6.10	0.11	23	1	43	1.1
	6	6.21	20	26.56	2.90	4.69	0.08	31	1	49	1.1
	7	8.20	19	32.93	3.97	6.37	0.13	29	1.1	52	1.2
	8	9.38	11	29.41	2.39	3.82	0.12	38	1	51	1
eSMEM	2	5.10	98	7.66	0.63	9.17	0.10	1	1	1	1
	3	5.01	95	9.38	0.68	7.33	0.08	4	1	0	0
	4	4.87	87	10.75	0.70	5.38	0.06	13	1	0	0
	5	5.28	92	12.07	0.81	4.95	0.05	8	1	0	0
	6	5.97	86	15.92	0.95	4.44	0.05	14	1.2	0	0
	7	7.92	75	19.72	1.24	4.54	0.04	25	1.1	0	0
	8	8.02	71	20.22	1.65	4.12	0.04	29	1.1	0	0

Table 4.8: Results comparison based on the overlapping synthesized models

	$\mathbf{M}$	τ_{ave} (%)	$N_{M'=M}$	$\frac{E_\alpha}{\alpha}$ (%)	E_μ	$\frac{E_\sigma}{\sigma}$ (%)	E_ρ	$N_{M'<M}$	$N_{f-,ave}$	$N_{M'>M}$	$N_{f+,ave}$
Zhang <i>et al.</i>	2	30.45	61	29.60	4.73	39.91	0.46	21	1	20	1.2
	3	35.56	58	29.95	5.41	39.51	0.30	25	1	17	1.4
	4	44.14	34	35.23	5.61	43.30	0.39	29	1.1	37	1.3
	5	43.12	22	42.29	9.85	35.59	0.35	35	1.2	43	1.6
	6	51.04	8	47.33	14.40	36.99	0.29	47	1.3	45	1.2
	7	59.10	13	35.46	13.84	39.62	0.30	39	1.4	48	1.3
	8	64.85	2	37.34	14.64	46.53	0.32	46	1.7	52	1.5
eSMEM	2	27.3	84	28.93	1.28	34.46	0.36	12	1	2	1
	3	29.97	72	28.86	1.99	38.43	0.31	23	1	4	1.1
	4	33.78	65	30.71	2.41	35.69	0.29	30	1.2	5	1.1
	5	39.13	64	32.52	4.73	39.07	0.32	29	1	7	1.3
	6	40.61	51	36.91	5.65	36.01	0.33	45	1.3	4	1
	7	44.38	45	38.27	6.37	38.81	0.29	55	1.4	0	0
	8	51.83	36	40.66	8.27	38.00	0.31	64	1.5	0	0

Table 4.9: Results comparison based on the overlapping synthesized models with Gaussian noise. ($PSNR = 31dB$)

4.8). In contrast, the eSMEM has more steady decrement in the order detection accuracy for higher model orders, and $N_{M'=M}$ is 71% for $\mathbf{M} = 8$. Although the accuracy can be reduced by the noise, the results show that the eSMEM has higher chance in detecting the correct number of components in a given mixture model, which means it is more accurate in detecting the number of protein particles in the cluster spots in real data. Furthermore, as shown in Table 4.9, the eSMEM algorithm has roughly similar false negative rates as the Zhang *et al.* SMEM algorithm and much less false positives as the Zhang *et al.* SMEM algorithm has the tendency to over-fit the models (higher $N_{M'>M}$ than $N_{M'<M}$) and hence cause false positive for the real data. The bias of eSMEM algorithm towards under-fitting is due to the prioritized split-merge selection, which means that the algorithm detect less spots in the image than actually exists, but the good side is that the false particle detection is eliminated and the overall model order selection accuracy is much higher. Hence, the eSMEM algorithm can be considered to be a conservative spot detection algorithm.

Position accuracy: Again, the positions of the mixture components in each model are indicated the means of the components. The accuracy of the positions is measured by the differences between the mean estimations and the ground-truth mean values, and hence is evaluated based on the average offset distance E_μ . The position accuracy is an important consideration for shape modelling as one of the goal in practice is to find the positions of the candidates in a mixture. E_μ is low for both algorithms on the non-overlapping models when the initial means are close

to the ground-truth. In contrast, E_μ becomes larger on the overlapping models as the ground-truth means are difficult to determine, and are needed to be estimated by the split-and-merge operations of the algorithms, and in this case, the eSMEM algorithm has smaller values of E_μ which are 67% less on average than the Zhang *et al.* SMEM algorithm. This means the eSMEM algorithm has higher accuracy in finding the positions of the mixture candidates and hence in estimating the position of the protein particles.

Size/shape accuracy: The co-localization analysis requires not only the positions of the protein particles in the image but also the size/shape information of the particles to measure the overlap between the different particles. For GMMs, the sizes of the components are indicated by the mixing weights α of the components where large weights are associated with larger components in the mixture, and the shape of a component relies on the covariance Σ of the component which is decomposed to σ_h , σ_k and ρ . The errors (e.g. $\frac{E_\alpha}{\alpha}$, $\frac{E_\sigma}{\sigma}$ and E_ρ) on the estimations of these parameters are low for both algorithms in the tests on non-overlapping models. This is because the data of each component is sufficient enough for the algorithms to estimate the optimal model parameters. However, when the components are overlapped, the parameter estimations become under-determined and optimized by the split-and-merge operation which usually converges to a local optimum, and hence the errors on the estimated parameters become larger. The accuracy on the parameter estimation relies on how close the modelling algorithm converges the parameters to the global optimum, which is equivalent to less estimation errors. As shown in Table 4.8, $\frac{E_\alpha}{\alpha}$ of the eSMEM algorithm is 63% less on average than the Zhang *et al.* SMEM algorithm, which means the eSMEM algorithm has higher accuracy in estimating the sizes of the mixture components, and hence estimating the sizes (or brightness) of the protein particles. Also, $\frac{E_\sigma}{\sigma}$ and E_ρ of the eSMEM algorithm are 4% and 46% respectively less on average than the Zhang *et al.* SMEM algorithm, and this means the eSMEM algorithm has higher accuracy in retrieving the shape information of the mixture components, and hence higher accuracy in estimating the shapes of the protein particles.

Noise sensitivity: In Table 4.7, the error of the estimated shape and parameters have high values when compared to Table 4.6 is due to the distortion of the synthesized model shapes caused by the noise. The noise also reduces the precision of both algorithms in detecting the correct order of the testing models, where some noise can trigger false positives in the component detection (see $N_{f+,ave}$), and in contrast some noise located between close components can lead to false negative detections (see $N_{f-,ave}$) by detecting the close components as one single component. Similar to the 2nd test set, the background noise in the 4th set with overlapping models reduces the accuracy of the two algorithms in shape fitting and true order detection, but since the eSMEM algorithm has a prioritized split-and-merge selection stage for the split and merge results which prevents over-fitting, the eSMEM algorithm has higher correct order detection rates and less false positive detection.

Furthermore, the sensitivities of the two algorithms for different background noise levels are also examined by fitting the non-overlapping and overlapping synthesized models with 6 different

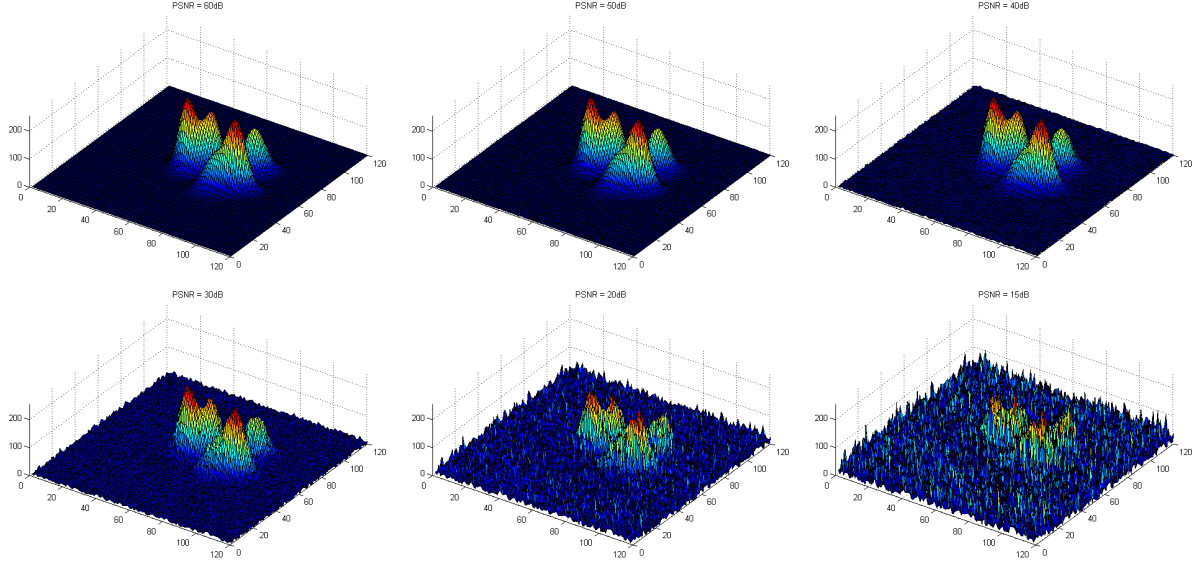


Figure 4.13: Samples of a synthetic model with 6 different background noise levels. (*This figure shows the appearance of a synthetic model when the Gaussian noise in 6 different levels are added to the image.*)

PSNRs which are 15dB, 20dB, 30dB, 40dB, 50dB and 60dB. The samples of a synthesized model with different background noise levels are shown in Fig. 4.13. The plots in Fig. 4.14 show the average error-signal-ratio τ_{ave} and the correct order detection rate $N_{M'=M}$ of the two algorithm for the synthesized models with $M=5$ in different background noise levels. The red lines indicates the results for the models with non-overlapping components and the blue lines are for the models with overlapping components. The solid lines indicates the results of the eSMEM algorithm while the dash lines are for the Zhang *et al.* SMEM algorithm. As expected, the accuracies of the algorithms increase (lower τ_{ave} and higher $N_{M'=M}$) when the noise level decreases. However, the Zhang *et al.* SMEM algorithm gave poor performance in the true order detection due to the stochastic split process, and hence the results of the algorithm given rough dash lines in Fig. 4.14(b). This shows the issue of the Zhang *et al.* SMEM algorithm in shape modelling when relying on random splits.

Stochastic vs deterministic: In order to visualize the comparison of the two modelling algorithms, the results of the two algorithm for a synthesized model in the 3^{rd} set are shown in Fig. 4.15. As shown in Fig. 4.15(a), the reference model is synthesized by combining three Gaussian models (i.e. $M = 3$) but has only one peak in the resultant shape, i.e. $M^{(0)} = 1$. Since the result of each split step in the Zhang *et al.* SMEM algorithm is selected from 20 random splits, the final result of the algorithm shown in Fig. 4.15(c) is the best split of the 20 proposals. The results of two other splits in the 20 proposals are shown in Fig. 4.16, which are less matched to the ground-truth.

As the Zhang *et al.* SMEM algorithm is not deterministic, the less matching splits can be

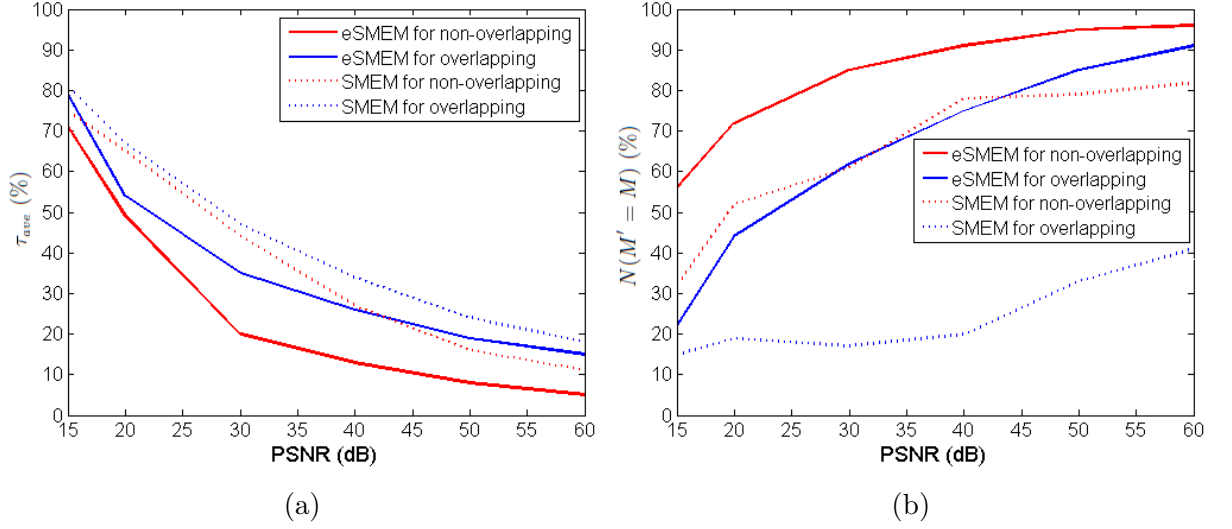


Figure 4.14: Plots of τ_{ave} and $N_{M'=M}$ of the eSMEM algorithm for $M=5$ with different background noise levels. The plots are derived from the results of the eSMEM algorithm for $M=5$ in both non-overlapping (the red lines) and overlapping (the blue lines) scenarios and 6 different background noise levels (where the PSNR is ranged in $[15, 60]$). (a) the average error-signal-ratio τ_{ave} of the eSMEM for different noise levels (b) the correct order detection rate $N_{M'=M}$ of the eSMEM for different noise levels.

the final results when repeating the modelling process. This randomness of the Zhang *et al.* SMEM algorithm is the key issue of the algorithm in shape modelling for the spot analysis. In contrast, the eSMEM algorithm is deterministic and as shown in Fig. 4.15(b), the results of the algorithm are similar to the Zhang *et al.* SMEM algorithm (the best of 20 splits), and the eSMEM algorithm has less fitting error and less over-fitting due to the prioritized split-and-merge selection in the model. Also, because of the deterministic splitting step, the observations show that the eSMEM process is 5 times faster on average than the Zhang *et al.* SMEM approach.

In the comparison, the modelling results from the four testing sets under different scenarios show that the eSMEM algorithm has higher accuracy in shape fitting, model order detection, position estimation and size/shape estimation than the Zhang *et al.* SMEM algorithm for the models with overlapping components and noise. This means the eSMEM is more likely to estimate the number of particle candidates correctly, and estimate the positions of the candidates correctly in the cluster spots in the real data, where the number and the positions of particles are the key factors for colocalization analysis. Also, the results show that the initialization is a key step for the eSMEM algorithm to detect the true model order and achieve good shape fitting when M is high, and the prioritized split-merge selection stage plays an important role in the algorithm to prevent over-fitting and to increase the chance of achieving the global optimum for the algorithm. Although eSMEM has greater model order accuracy, it is biased towards under-fitting due to the prioritized split-merge selection whereas the Zhang Method is

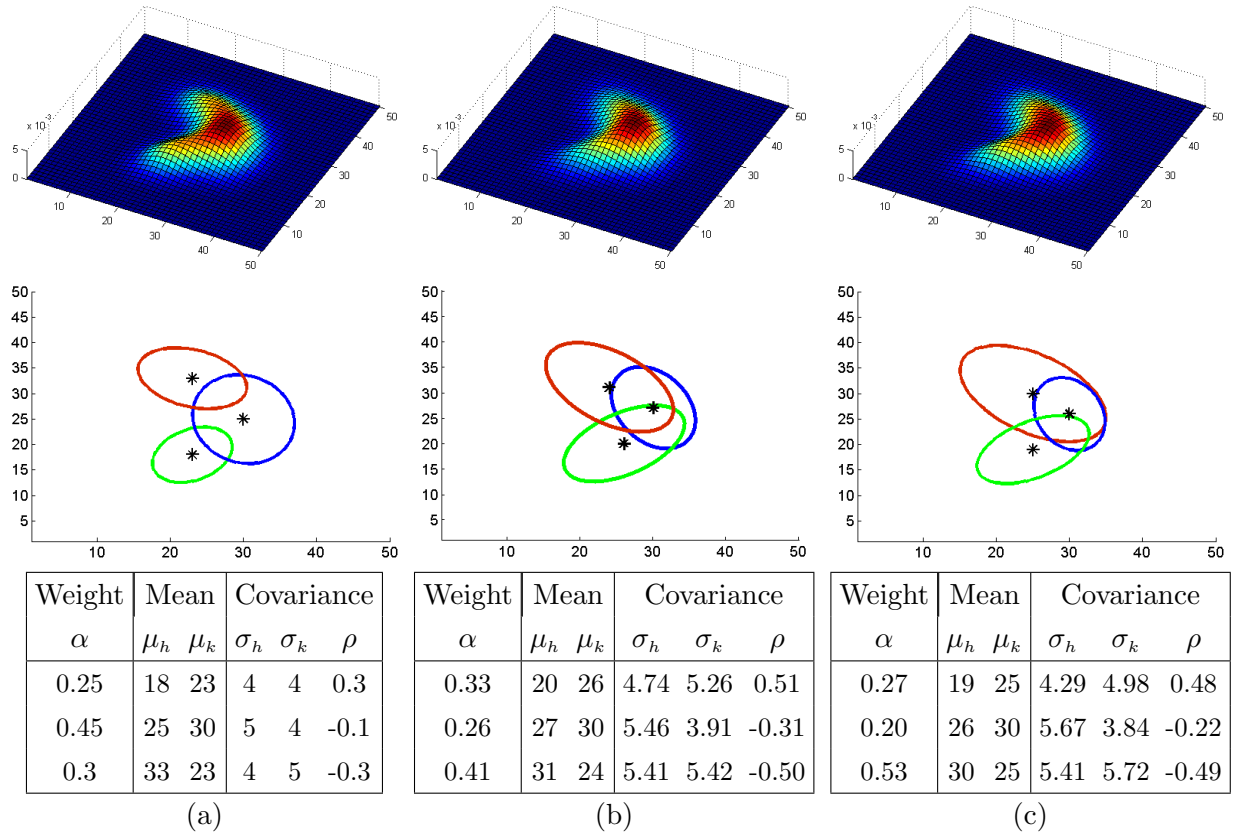


Figure 4.15: Modelling results of the eSMEM algorithm and the Zhang *et al.* SMEM algorithm. (a) a synthetic GMM, i.e. the target shape. (b) shows the estimation of (a) using the eSMEM algorithm, the error-to-signal ratio τ is 8.97%. (c) the best estimate of the Zhang *et al.* SMEM algorithm from 20 random splits on (a), the error-to-signal ratio is 12.37%. The top row shows the profile of the mixture models and middle row shows the corresponding contours of the components in the mixture models. Note that the black (*) are the mean of the components. The last row shows the parameters for corresponding mixture models.

balanced. This means that eSMEM will underestimate the number of spots in an image, but at the same time, the false positive detection is eliminated in the eSMEM modelling process and therefore the algorithm can be considered to be a conservative detection algorithm that gives increased robustness (higher model order selection accuracy) over the Zhang *et al.* method. The results of Spotnick on real data are shown and discussed in the next section.

4.3 Results on Real Data

Recall that the input image is firstly decomposed into small patches of spot-regions, and then modelled by fitting a Gaussian mixture to the surface of each isolated spot-region. The results of the modelling operation on some practical samples are given in Fig. 4.18, 4.19, 4.20 & 4.21, where

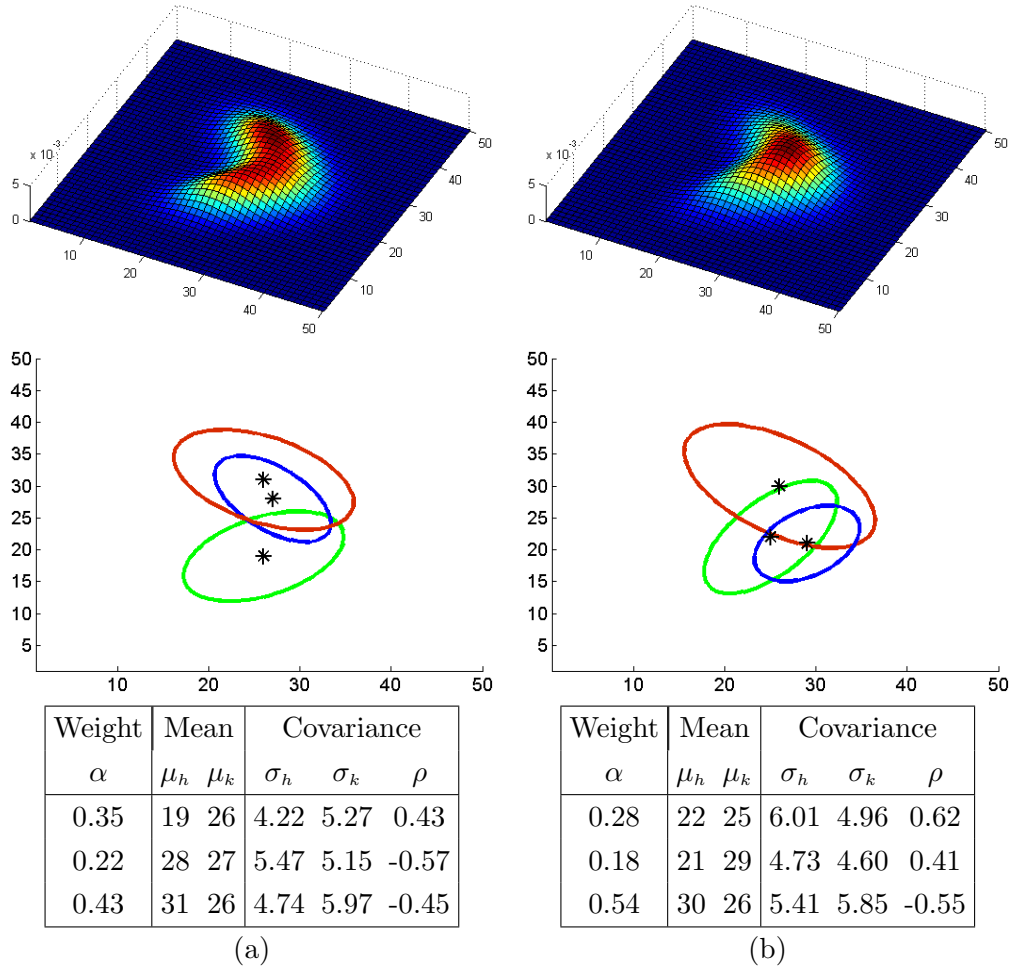


Figure 4.16: Variation in the results of the Zhang *et al.* split method. (a) & (b) are two different estimates of the Zhang *et al.* SMEM algorithm on the same input shown in Fig. 4.15(a), and the error-to-signal ratios are 13.42% and 14.62% respectively. The top row shows the profile of the mixture models and the middle shows the corresponding contours of the components in the mixture models. Note that the black (*) are the mean of the components. The last row shows the parameters for corresponding mixture models.

the samples are extracted from a given image in Fig. 4.17. In each figure of the samples, the first row shows the extracted spot-regions. The second row shows the local background estimates of the samples. The third row shows the pre-processed (background subtracted) spot-regions. Finally, the last row shows the estimated models for the samples. Note that the spot-regions shown in Fig. 4.19 and Fig. 4.18 are extracted from a brighter region and a darker region respectively in the given image. Since the variation in background luminance has been solved by the contrast modification method in Section 4.1.2.1, the modelling results for the spots in different regions of the given image are, as shown Fig. 4.18 & 4.19, not affected by the brightness variation.

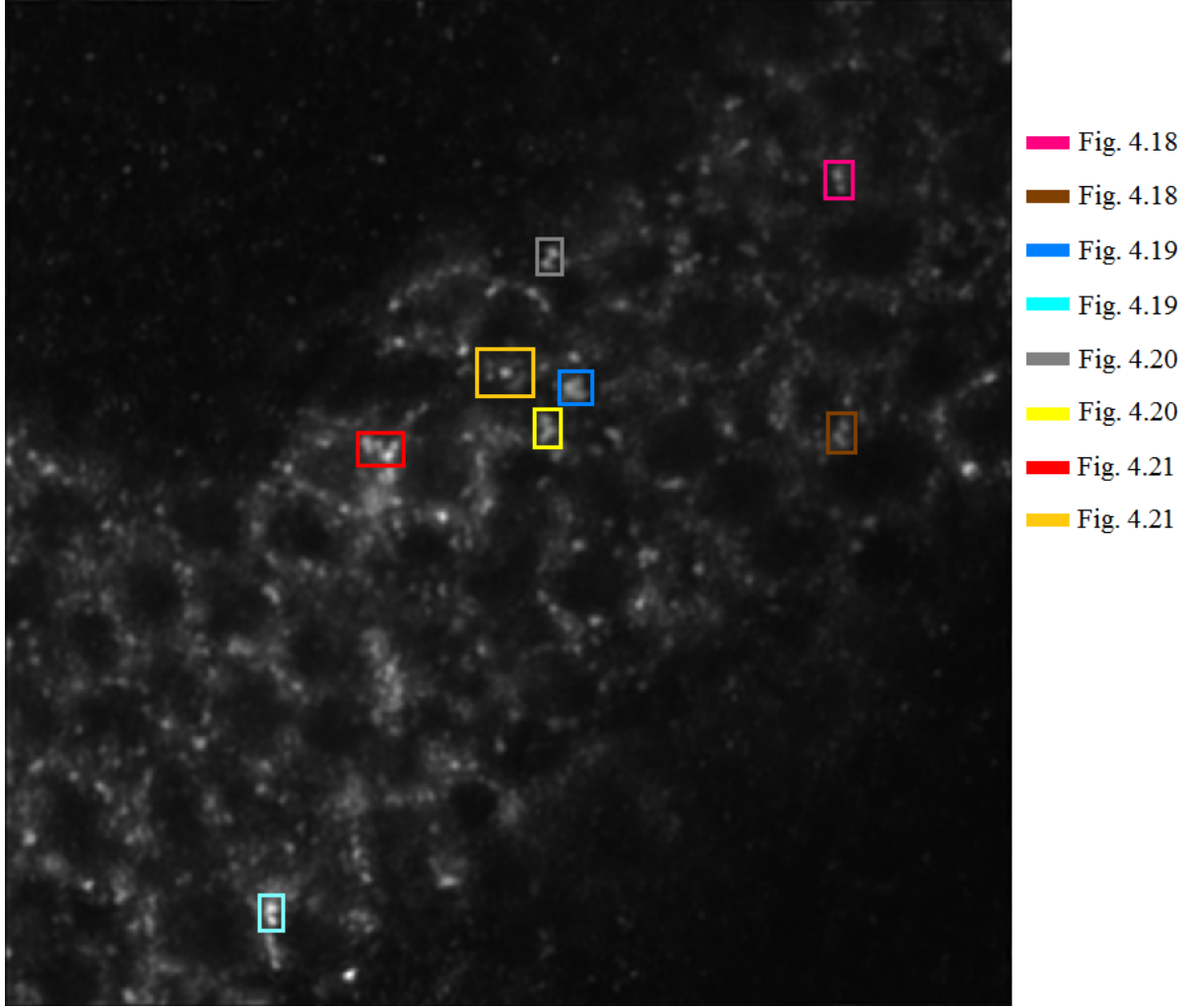


Figure 4.17: De-noised original image *with color-boxed sample spot-regions being modeled and shown in the following figures.*

As shown in the results on real data, the eSMEM modelling algorithm is successful in characterizing spot shapes analytically on the spot patches. Even when protein particles are clustered and yield merged spot shapes, the algorithm can generate reliable estimation of the sizes and positions of particles in those spots.

However, the estimated shapes are less well matched to the real spot samples when compared to the artificial data. This is caused by two inherent problems in the pre-process stage of the modelling process. The first problem arises from the operation of spot-region extraction. As the region boundary is defined by the zero-crossing method along with the DoG technique, the shape data that are outside of the boundary would be omitted from the complete profile of a spot-region. In other words, the selected area for modelling does not cover the complete area of the actual spot-region. In most situations, even if the region is incomplete, enough data will be available to estimate the model parameters accurately. As shown in Fig. 4.6(c), the data inside

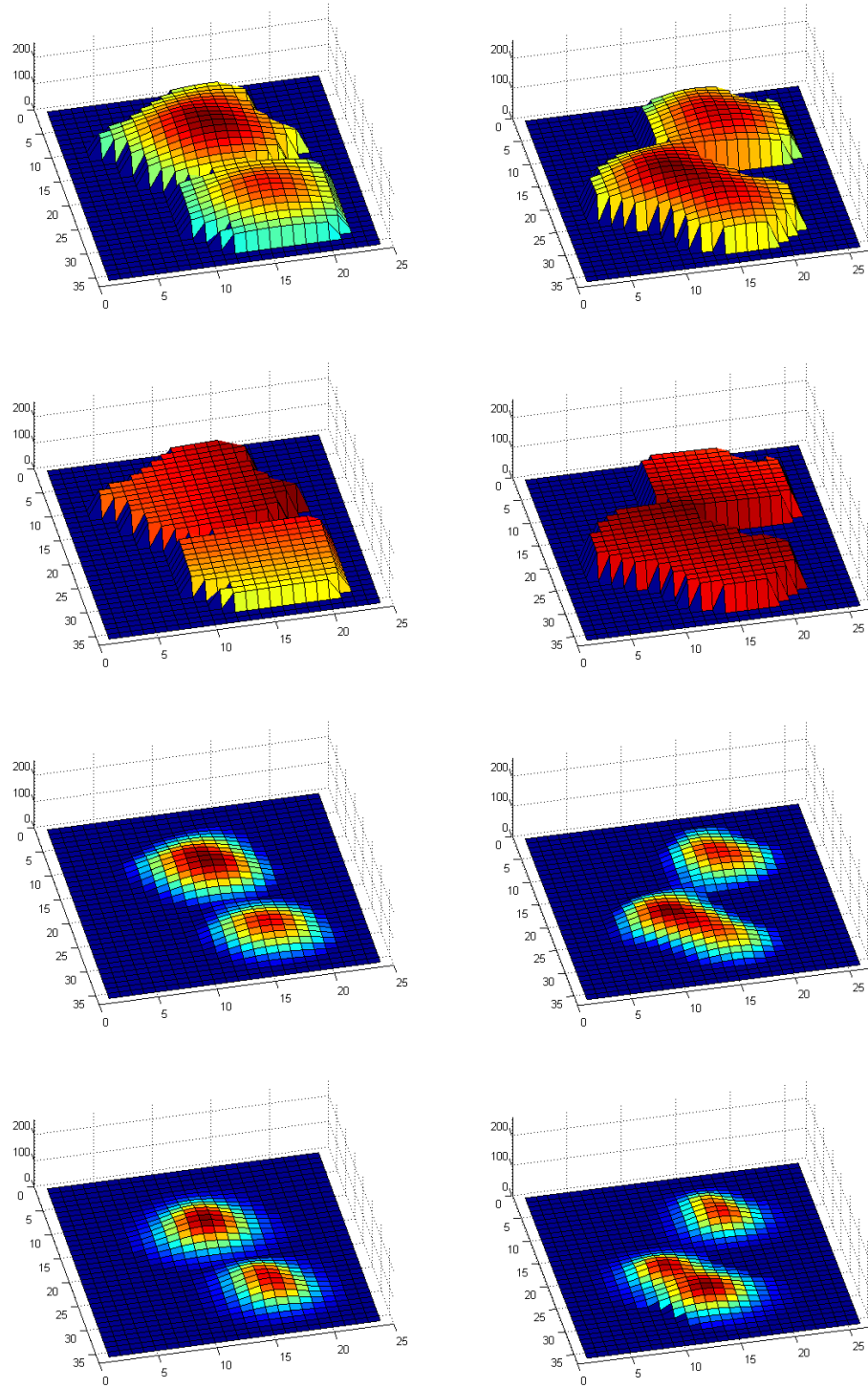


Figure 4.18: Practical results for shape modelling I (*First row shows the profile of spot-regions extracted from Fig. 4.17; second row shows the profile of local background estimates of the spot-regions; third row shows the pre-processed (background subtracted) profiles; bottom row shows the estimated mixture models of the objective spot-region profiles using eSMEM shape modelling*)

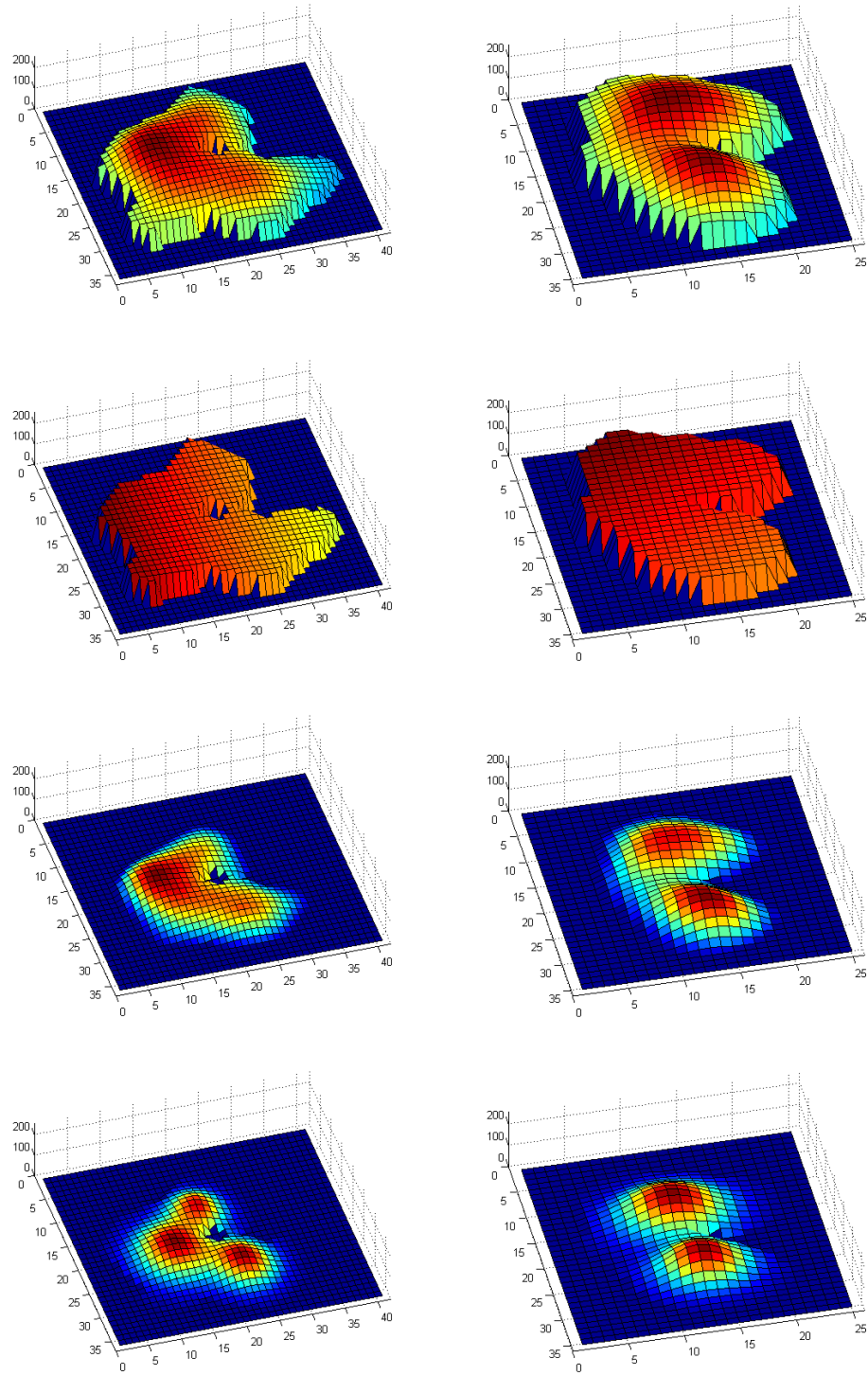


Figure 4.19: Practical results for shape modelling II (*First row shows the profile of spot-regions extracted from Fig. 4.17; second row shows the profile of local background estimates of the spot-regions; third row shows the pre-processed (background subtracted) profiles; bottom row shows the estimated mixture models of the objective spot-region profiles using eSMEM shape modelling*)

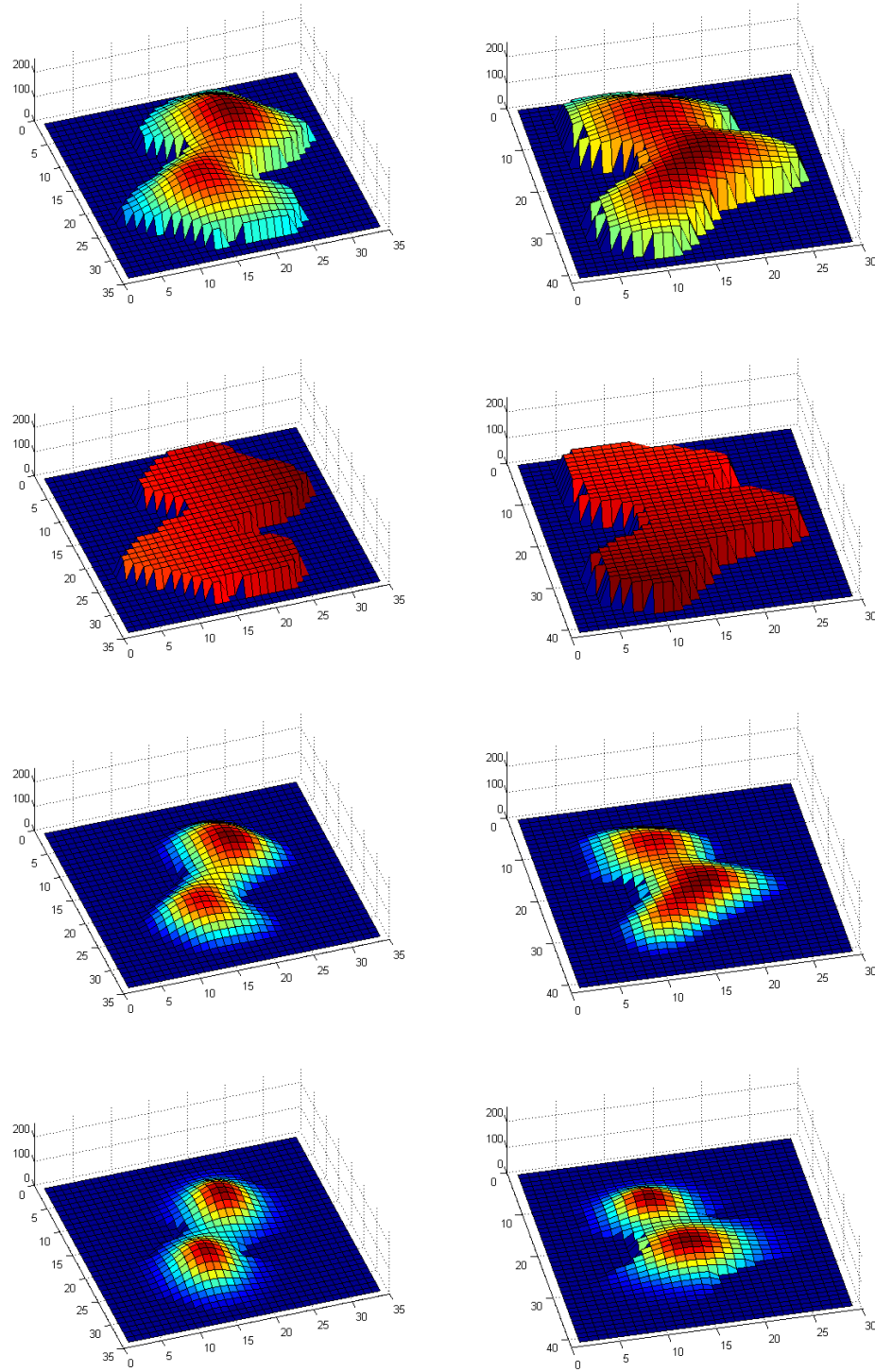


Figure 4.20: Practical results for shape modelling III (First row shows the profile of spot-regions extracted from Fig. 4.17; second row shows the profile of local background estimates of the spot-regions; third row shows the pre-processed (background subtracted) profiles; bottom row shows the estimated mixture models of the objective spot-region profiles using eSMEM shape modelling)

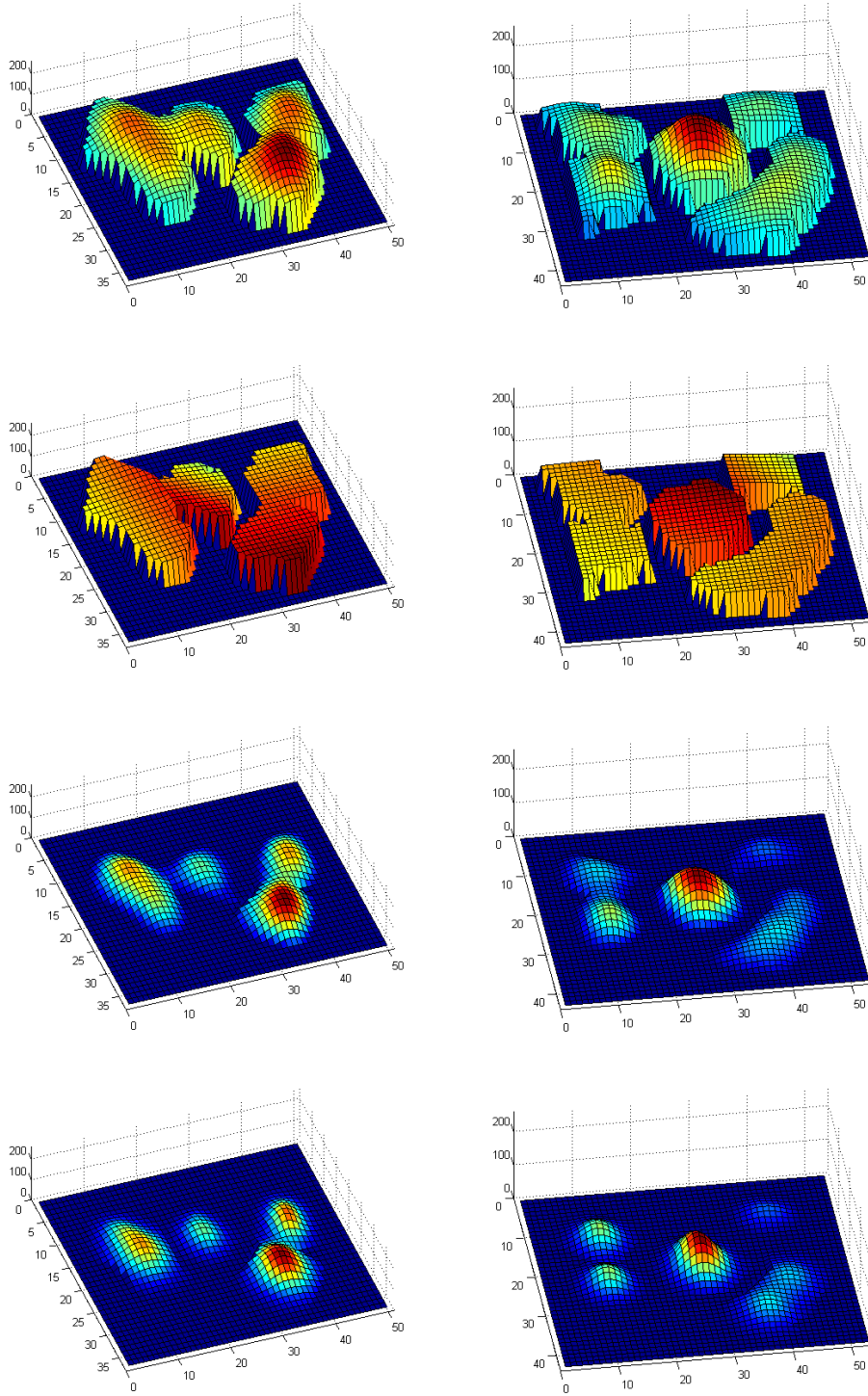


Figure 4.21: Practical results for shape modelling IV (*First row shows the profile of spot-regions extracted from Fig. 4.17; second row shows the profile of local background estimates of the spot-regions; third row shows the pre-processed (background subtracted) profiles; bottom row shows the estimated mixture models of the objective spot-region profiles using eSMEM shape modelling*)

the boundary defined by the DoG technique is sufficient for the GMM shape fitting. However, in general, it is possible that the modelling accuracy could be reduced. The incomplete region can trigger the other problem in the local background estimation process.

Recall that the background estimation is based on the assumption of local background proposed in *Section 4.1.3*. This assumption states that while the actual intensity of a spot is negligible at the boundary when compared to the background intensity and hence the boundary spot intensity values are reasonably assumed as zero. Therefore, the boundary of a selected spot-region can be considered as the local background data. However, if the segmented spot region is smaller than the true spot shape, then the spot intensity will not be zeros at the boundary (see Fig. 4.22). Consequently, this defect reduces the fit quality which is shown as the blue line in Fig. 4.22(c). However, since the biological research is still in the study stage, there is no ground-truth available for the real data as the actual characteristics of the protein particles are still uncertain. Hence, the effect of the two issues can not be quantitatively evaluated, but instead are evaluated based on qualitative interpretation. For instance, as shown in the third rows in Fig. 4.18-Fig. 4.21, the local background subtraction removes the step continuity from the exacted real data in the first rows.

Furthermore, the goal of this task is to measure the co-localization of the different proteins, which is to measure the overlapping of the different protein particles from two superimposed images. The co-localization analysis requires the number, positions and the approximate size of the candidate particles in the spots in the image, but not the exact shape of the spots. Although the defects in the pre-processing stage would affect the accuracy in shape estimation of the spots, they would not affect the model order detection and have little effect in the mean estimation for the mixture components. Furthermore, as shown in Fig. 4.19, the original spot-region with irregular shape appearance Fig. 4.19(a) is modelled by a Gaussian mixture with three components Fig. 4.19(b). Each component locates the corresponding particle candidate in the patch, and associated Gaussian parameters characterize the size and shape of that particle. This means that Spotnick is a sufficient algorithm in characterizing the candidate particles in the cluster spots for the co-localization analysis in this task.

4.4 Data Analysis

The goal of the memory formation that this work supports is to investigate the existence of specific substances, where a specific substance is located by the co-localized proteins of different types. The investigation is based on the co-localization of the different proteins. This section discuss the ways in which the raw data generated by Spotnick has been distilled to make it more useful for biological analysis. The mixture parameters generated by Spotnick are attributed to protein particles in the test image and these parameters are used to drive a co-localization analysis of protein particles in the two registered images. This approach has been applied by the researchers in neuroscience to the study of long-term memory formation [46].

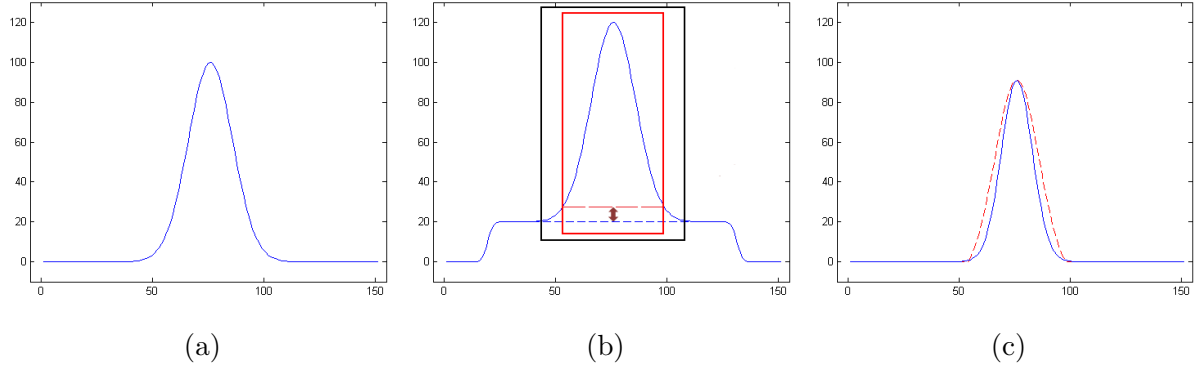


Figure 4.22: Defect in background estimation. (a) Target model; (b) The region in red box is typically extracted instead of the black one (the ideal one) for shape modelling; (c) The red line indicates the extracted sample in red box in (b) after background subtraction and the blue line represents the estimated model of the red line.

To measure the co-localization of the different protein particles from two superimposed images, the number, sizes and shapes of the candidate particles in each spot in the images are parameterized by the corresponding GMM estimated by Spotnick. Since the spot patches that are segmented from the given image are modelled by the eSMEM algorithm separately, the estimated models are combined to form a complete estimation for the input image. However, the modelling estimation of the spot patches is based on the normalized intensity of the image. For comparison with the original image gray-scale, the modelling estimates are adjusted back to the gray-scale by following procedure

$$\hat{\mathbf{I}}_{EM} = \mathbf{I}_{EM} \sum \mathbf{I} \quad (4.5)$$

where $\mathbf{I}$ is the intensity profile of the original spot patch before normalization. Moreover, since the ‘rescaling’ factor $\sum \mathbf{I}$ is the same for all mixture components from the same modelling estimate, for data storage purpose, the intensity adjustment is applied to the mixing weight α and the ‘rescaled’ mixing weight $\hat{\alpha}$ is defined as

$$\hat{\alpha} = \alpha \sum \mathbf{I} \quad (4.6)$$

The estimated GMMs after the intensity adjustment are then combined in the distribution respect to the corresponding extracted spot patches. As shown in Fig. 4.23 and Fig. 4.24, the estimations of the individual spot patches are combined into a complete estimation of the sample images. However, the combined image estimate does not include the background information when the original image contains irregular background brightness. In order to enable the user to compare the estimate with the original image, the combined image estimate is added with the local background estimates associated to the corresponding spot estimates. Two examples of the background-added image estimates are given in Fig. 4.23 and Fig. 4.24.

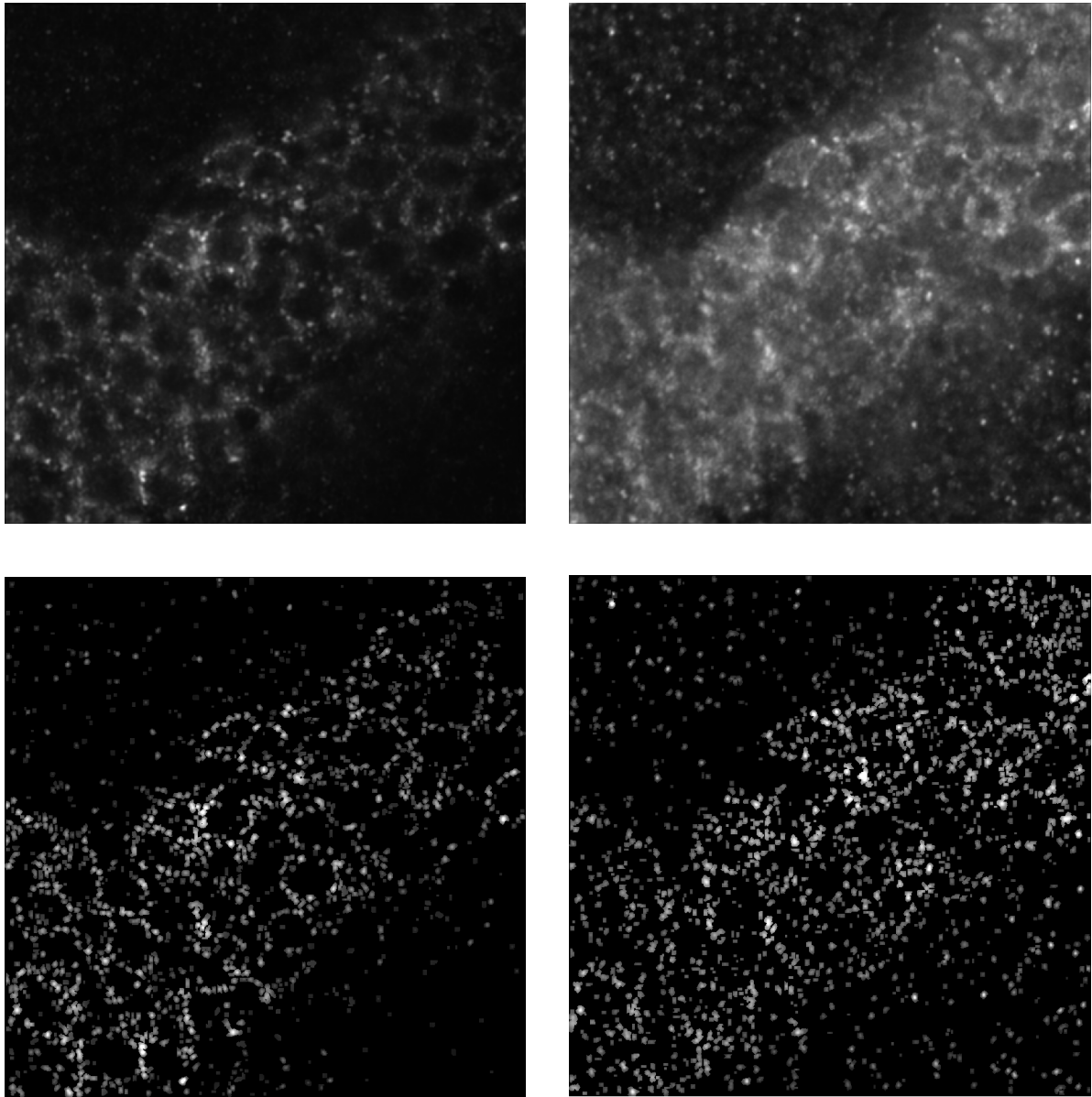


Figure 4.23: Sample of synthesized images I (*The left column is from the red channel of a sample double-stained image and the correct column is from the green channel of the sample image. The top row shows the de-noised original gray images where the bottom row shows the corresponding synthesized images with the estimated spot-region models.*)

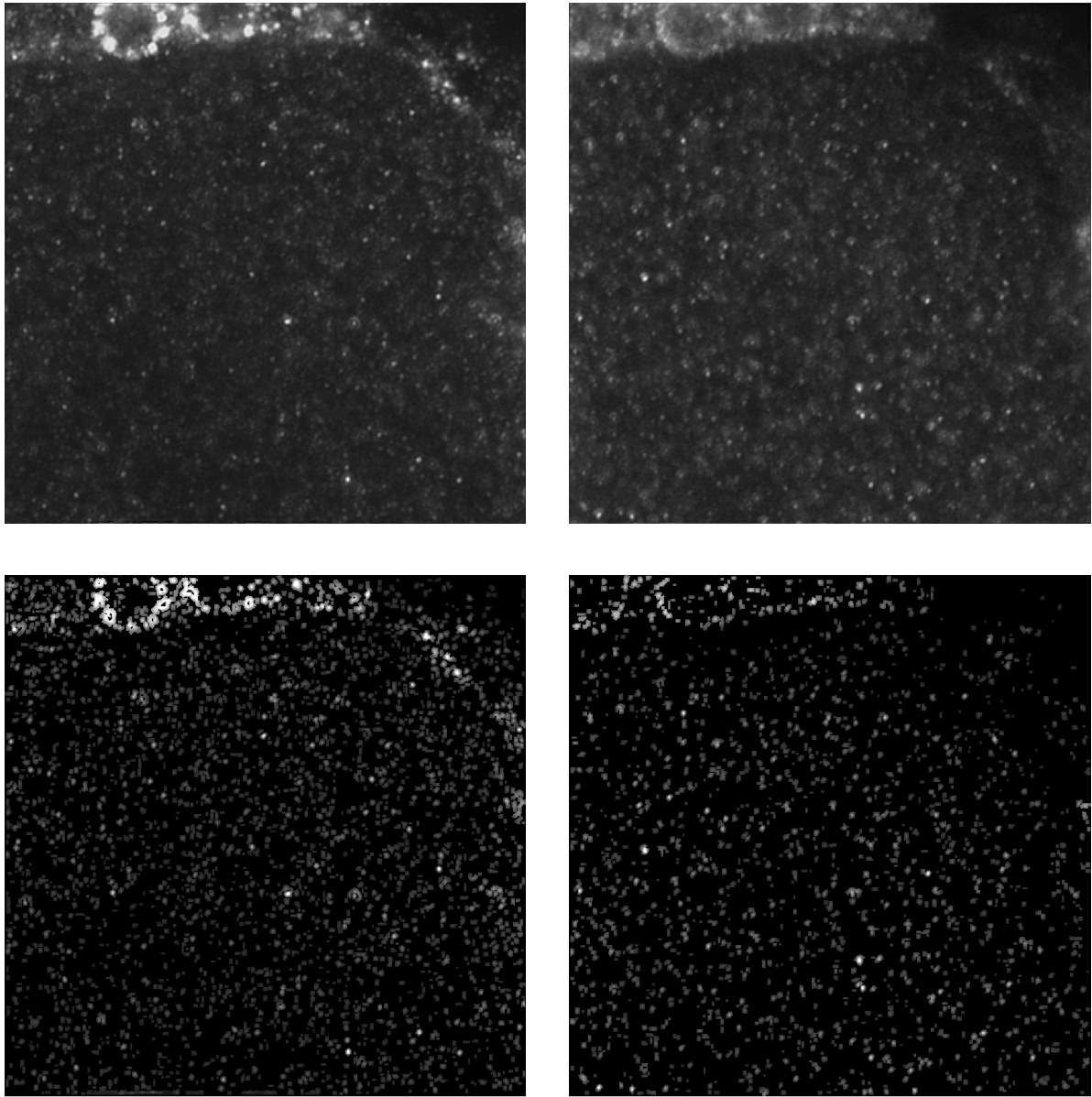


Figure 4.24: Sample of synthesized images II (*The left column is from the red channel of a sample double-stained image and the correct column is from the green channel of the sample image. The top row shows the de-noised original gray images where the bottom row shows the corresponding synthesized images with the estimated spot-region models.*)

4.4.1 Parameter Registration and Statistics

For data storage and analysis, the spot estimates are registered in a single parameter set for the given image where the parameters of each individual spot patch are distinguishable by the corresponding numeric labels of the patches. An example for the left image in Fig. 4.23 is given in Table 4.10, where each row of the data set consists of the parameters of a Gaussian mixture component. Recall that each spot in the original image is actually a cluster of protein particles. When the spot is fitted by a GMM, each mixture component is the particle estimate of the corresponding candidate protein particles in the cluster spot. Hence, the parameters in each row of the data set are also the estimate of the corresponding protein particle in the image.

In order to analyze the geometric information of the spots, three histograms are computed from the parameter set. These histograms, as shown in Fig. 4.25, offer statistical information on different geometric aspects of the protein particles in the given image. The first histogram is about the size of the protein particles. Since the shape of each mixture component in the spot estimates is controlled by the covariance matrices $\Sigma = \begin{bmatrix} \sigma_h^2 & \rho\sigma_h\sigma_k \\ \rho\sigma_h\sigma_k & \sigma_k^2 \end{bmatrix}$, by assuming the protein particles are circular, the size of each particle can be roughly represented by the ‘radius’ $\sigma = \frac{1}{2}(\sigma_h + \sigma_k)$ of the corresponding mixture component. The plot in Fig. 4.25(a) is a histogram of σ which is equivalent the histogram of size of the protein particles in the given image. The next plot Fig. 4.25(b) is a histogram of $\hat{\alpha}$ of the mixture components. When a spot profile is the intensity of a cluster of protein particles, the ‘rescaled’ mixing weight $\hat{\alpha}$ can be considered as the intensity estimate of the corresponding particles in the image. Therefore, Fig. 4.25(b) is histogram of the particle intensities. Finally, Fig. 4.25(c) is the histogram of the local background brightness of the particles in the given image that are represented by the local background estimates $B_{\vec{\mu}}$. The three histograms allow the users to draw brief conclusions from the estimation about the sizes, brightness of the particles as well as the background brightness of where the particles located. These histograms enable the users to select the spots with the desired characteristics by setting the ranges in interest on the histograms.

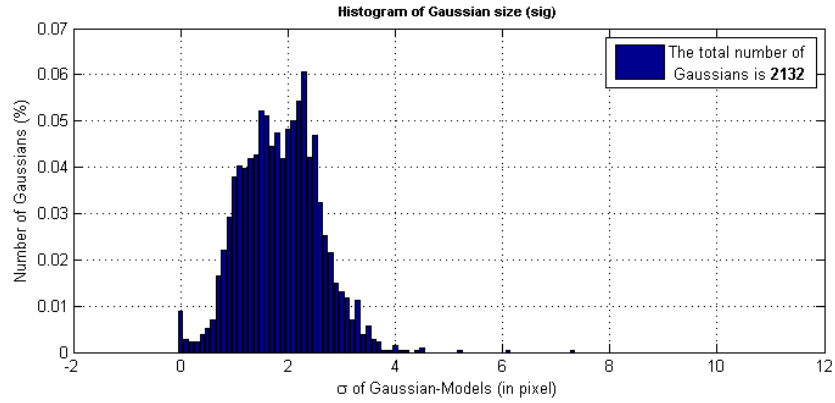
4.4.2 Co-localization Analysis

The primary task of this work is to analyse the co-localization of the protein particles in two different images. Since the positions (μ) and the sizes (σ) of the particles have been estimated as μ and σ of the Gaussian mixture components in the shape modelling process, the objective is simply to find the closest Gaussian component in the second image for each spot in the present image with respect to their sizes, and vice versa. This is similar to a ‘nearest neighbour’ problem that finds the nearest point to the current point but in this case, the searching process is performed in the other image instead of within a single image. Hence, this searching process is named as cross-nearest-neighbour measurement.

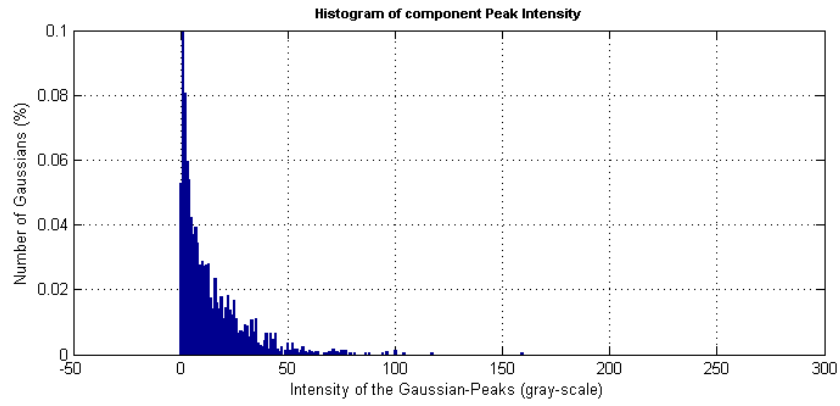
Suppose there are Q particles in the ‘red’ image with ‘mean’ positions $\vec{\mu}^{red}$, and V particles

Spot #	h_{start}	h_{end}	k_{start}	k_{end}	$\hat{\alpha}$	μ_h	μ_k	σ_h	σ_k	ρ	$B_{\vec{\mu}}$
1	68	76	4	10	8.7636	6	4	1.4121	1.4687	0.049	28.6236
1	68	76	4	10	5.2385	3	4	0.9089	1.5837	0.0667	28.6236
2	151	163	4	14	23.8952	7	5	2.3107	1.9632	0.0104	0
2	151	163	4	14	0.2713	3	6	0.566	1.2571	0.0112	0
2	151	163	4	14	1.9349	4	5	1.1429	1.7103	-0.0564	0
2	151	163	4	14	3.4416	4	5	1.0769	1.7687	-0.0554	0
3	194	206	4	8	10.0159	8	2	1.5022	1.1951	-0.1683	31.357
3	194	206	4	8	4.848	5	3	1.3553	1.1137	-0.3094	31.357
3	194	206	4	8	2.89	5	4	1.4304	1.211	-0.2074	31.357
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
1099	32	47	1011	1019	12.137	9	5	2.5649	1.8986	0.2041	0
1099	32	47	1011	1019	4.4085	6	4	1.2231	1.8771	-0.1453	0
1099	32	47	1011	1019	0.1779	5	4	0.6671	1.9557	-0.1289	0
1099	32	47	1011	1019	0.2843	5	4	0.6695	1.861	-0.1219	0
1100	458	464	1015	1021	5.2195	4	4	1.6542	1.1883	0.1293	44.1422
1100	458	464	1015	1021	0.3047	2	4	0.6454	1.1156	-0.0489	44.1422
1100	458	464	1015	1021	0.3002	2	4	0.6448	1.1265	-0.0509	44.1422

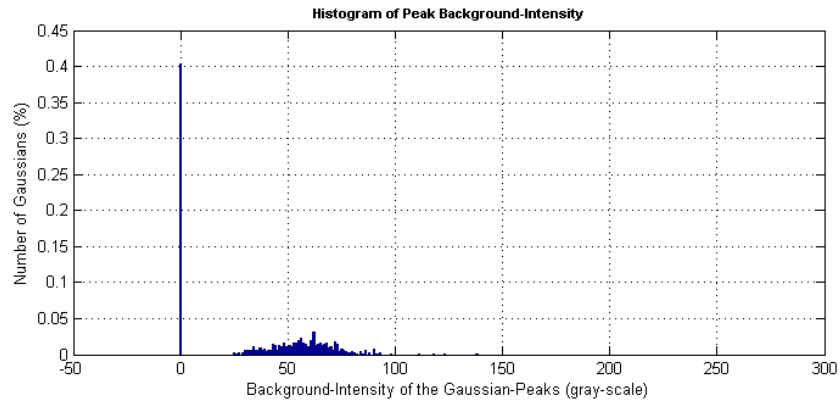
Table 4.10: An example of parameter set for modelling the left image in Fig. 4.23 (Each row of the table consists of the parameters of a Gaussian mixture component. The first column, ‘Spot #’, contains the numeric labels of the ‘parent’ spots that the mixture components are estimated from. These labels are assigned to the spots in the labelling step of the pre-processing stage stated in Section 4.1.2.2. The next four columns consist of the location and size information of the spot patches in the given image, this information is represented by starting coordinates and the ending coordinates $(h_{start}, k_{start}) \rightarrow (h_{end}, k_{end})$ respectively in the image. The column with $\hat{\alpha}$ consists of the ‘rescaled’ mixing weights. The next two columns represent the mean $\vec{\mu} = [\mu_h \mu_k]$ of the components in the corresponding spot patches. The following three columns store the information of the covariance of the components where $\Sigma = \begin{bmatrix} \sigma_h^2 & \rho\sigma_h\sigma_k \\ \rho\sigma_h\sigma_k & \sigma_k^2 \end{bmatrix}$. The last column consists of the estimated local background values B at corresponding mean positions $\vec{\mu}$ of the components which are then denoted as $B_{\vec{\mu}}$. The local background values B is an extra information for further data analysis.)



(a)



(b)



(c)

Figure 4.25: Analytical histograms (The top is a histogram of the standard deviations σ of the all the estimated Gaussian components of a given image, where σ also be considered as the sizes of the protein particles in a microscopical image. The middle is a histogram of the intensity values at the mean positions, also named as the peaks, of the estimated Gaussian components of the given image. The bottom is a histogram of estimated background values at the peaks of the Gaussian components.)

in the ‘green’ image with ‘mean’ positions $\vec{\mu}^{green}$, the process of finding the nearest neighbour for the particles q and v in the ‘red’ and ‘green’ image respectively is described as follows

$$\begin{aligned} \mathbb{N}_q^{red} &= \arg \min_v |\vec{\mu}_q^{red} - \vec{\mu}_v^{green}| \\ \mathbb{N}_v^{green} &= \arg \min_q |\vec{\mu}_v^{green} - \vec{\mu}_q^{red}| \end{aligned} \quad (4.7)$$

where $\mathbb{N}_q^{red}$ denotes the nearest particle in the ‘green’ image for particle q in the ‘red’ image, and conversely, $\mathbb{N}_v^{green}$ denotes the nearest particle in the ‘red’ image for particle v in the ‘green’ image.

To determine whether the correlation between the locations of the particles in the two channels is significant, a control experiment is needed. This control is created by measuring the cross-nearest-neighbour of particles with the same amount of particles in two images but uniformly randomly distributed. Hence, a cumulative histogram of the estimated nearest-neighbour distances is compared to the control of artificially generated data.

After finding the nearest neighbour for all the particles, a cumulative histogram of the distances (R) between the particles and their nearest neighbours is computed for each image. The cumulative histograms are shown (in ‘red’ with ‘blue’ dots) in the first row of Fig. 4.26, where Fig. 4.26(a) is in respect of distance of the nearest neighbour for the particles in the ‘red’ image, and Fig. 4.26(b) is for the particles in the ‘green’ image.

The generation of the control cumulative histogram is based the cross-nearest-neighbour measurement for uniformly randomly distributed points. The control cumulative histograms are shown as the ‘green’ plots in Fig. 4.26(a)&(b). The cross-nearest-neighbour search for random distribution is performed between two images, the ‘r’ image and the ‘g’ image. The ‘r’ image contains Q points and the ‘green’ image contains V points, so that the two images have the same number of points as the amount of particle estimates in the real images. In contrast with the real data, the points are distributed uniformly randomly in the artificial ‘red’ and ‘green’ images. The cross-nearest-neighbour measurement of the points among the artificial images is estimated in the same manner as the real data. As shown in Fig. 4.26(a)&(b), when the real data plots (in ‘red’ with ‘blue’ dots) are closer to the randomly-distributed measurement (in ‘green’), the actual protein particles are more likely randomly distributed in the given image. Otherwise, the particles are distributed in a certain pattern which is of interest in the biological context of this work.

Note that the co-localization of two particles depends not only on the distance between the particles but also the sizes of the particles. After determining the distribution status of particles in the given image, the co-localization analysis of the particles is computed by calculating the ratio of the distance (R) between a particle q (or v) and the associated cross-nearest-neighbour $\mathbb{N}_q^{red}$ (or $\mathbb{N}_v^{green}$) to the sizes of the two particles. This cumulative histogram allows users to determine the significance of co-localization between the particles based on the user defined co-localization ratio (distance/radius ratio). Since σ are assumed to represent the radius of the circular protein particles, the co-localization ratio is then calculated by $R/(\sigma_r + \sigma_g)$. The

cumulative histograms of co-localization ratio $\frac{R}{(\sigma_r + \sigma_g)}$ for both channels are shown in the second row of Fig. 4.26. Again, the plot in Fig. 4.26(c) is for the particles in the red-channel and Fig. 4.26(d) is for the green-channel. Furthermore, some tools are developed for the biologists to use Spotnick for parameterizing the protein particles and perform co-localization analysis, and are illustrated in *Appendix B*.

4.5 Conclusion

This chapter has shown how the eSMEM algorithm introduced previously, can be employed to perform spot modelling and analysis on real biological data. The key addition is the introduction of the pre-processing stage to overcome the scanning-noise and the background luminance problems. In order to smooth the stripe-like scanning noise induced by the microscope, the first step in the pre-processing stage is noise reduction using the Wiener deconvolution. The second step is to segment the separated spot-patches from the image for the purpose of reducing the computational load for shape modelling. However, the segmentation is affected by the irregular background brightness inherent in the image. A contrast enhancement method is applied so that the spot segmentation becomes approximately invariant to the background luminance. In order to achieve the optimal shape modelling result, the last step in the pre-processing subtracts the background intensity from the segmented spot-patches by estimating the backgrounds as 2D first-order polynomials.

The complete shape modelling algorithm, including both the pre-processing and the eSMEM algorithm, is named ‘Spotnick’. The shape modelling compatibility and the performance of the split-and-merge operation of Spotnick are firstly examined by various synthetic testing models, and Spotnick gave good modelling results. Also, the eSMEM algorithm is compared with the Zhang *et al.* SMEM algorithm on synthetic Gaussian mixture models with non-overlapping or overlapping components in different model orders. Since the real data has background noise, the two algorithms are also tested by the synthetic models with Gaussian background noise in the images of the models. The comparison shows that the eSMEM algorithm has better performance in shape fitting, model order detection, position estimation and size/shape estimation, and is outstanding for larger number of mixture components. The results also show that the importance of the prioritized split-merge selection stage for the eSMEM algorithm to prevent over-fitting for the spot analysis, and also shows that the proposed initialization method helps the algorithm to achieve better modelling results.

On real data, Spotnick provides approximations to the ‘merged’ spots and is sufficient to parameterize the candidate particles in the cluster spots for the co-localization analysis. After showing the shape modelling results, the usefulness of the results to the biological analysis is demonstrated by showing how the shape models are used to perform data analysis and co-localization analysis on the protein particles in the images. In reality, the tools of Spotnick (see *Appendix B*) are being utilized in the neuro-study of memory formation for parameterizing the

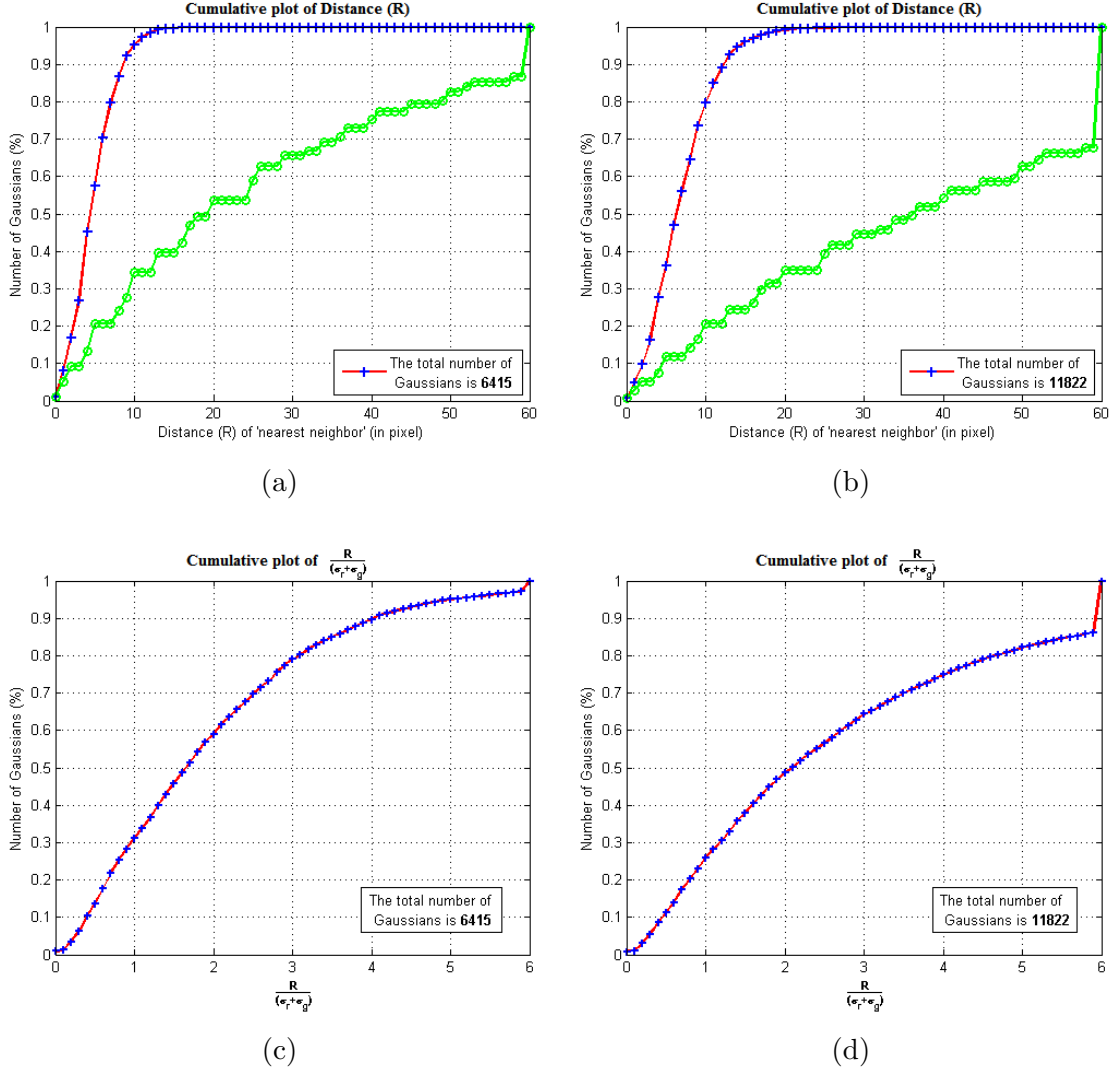


Figure 4.26: Sample of Co-localization analysis for images in Fig. 4.23 (The first row shows the cumulative plots with the number of particles (%) versus the distance (in pixels) between a particle in one image and its nearest neighbour in the other image. (a) is in respect of the particles in red-channel with neighbours in green-channel, and (b) is for the particles in green-channel with neighbours in the red-channel. The 'green' lines indicate the control cumulative histograms for the case of the same number of particles but randomly distributed in the given images. (c) and (d) are the cumulative histograms of $R/(\sigma_r + \sigma_g)$ for the red and green channel respectively.)

protein particles and performing co-localization analysis, and has already made contribution to the biological study. The biologists have found early observations and characterization of candidate synaptic mRNP particles that contains Me31B, and have suggested a simple model in which Me31B mediates the formation of synaptic mRNP particles that contain locally repressed mRNAs in [46].

5

Synapcut: a wavelet-based Bayesian framework for synapse segmentation ¹

The formation of memory is governed by protein synthesis. Since the synthesis takes place in the synapses of the neuron cells, the shape information of the synapses for where the protein particles are located during protein synthesis is important for the study of memory formation. However, the shape information is imaged as a series of slices of the synapses at different depths of the specimen. The 3D visualization of the synapses would require segmentation of the synaptic shape from the stacks of image slices. Similar to the protein particles in *Chapter 3*, the synapses are also labelled with fluorescent markers in the living specimen. However, the observed images usually appear with irregular brightness due to the uneven distribution of the fluorescent particles and the different marker concentrations in the background. At the same time, the images have out-of-focus objects caused by the detection of out-of-focus signal from other depths of synapse structure. To distinguish the in-focus objects with various luminance from the out-of-focus objects, the segmentation of the synapses from the image stacks is heavily reliant on manual analysis. In this chapter, a novel Bayesian model, that uses ‘wavelet features’ and Laplacian object frames, is introduced for synapse segmentation in given image stacks, and hence creates 3-D models of the synapses.

¹This work is presented in part in [76]

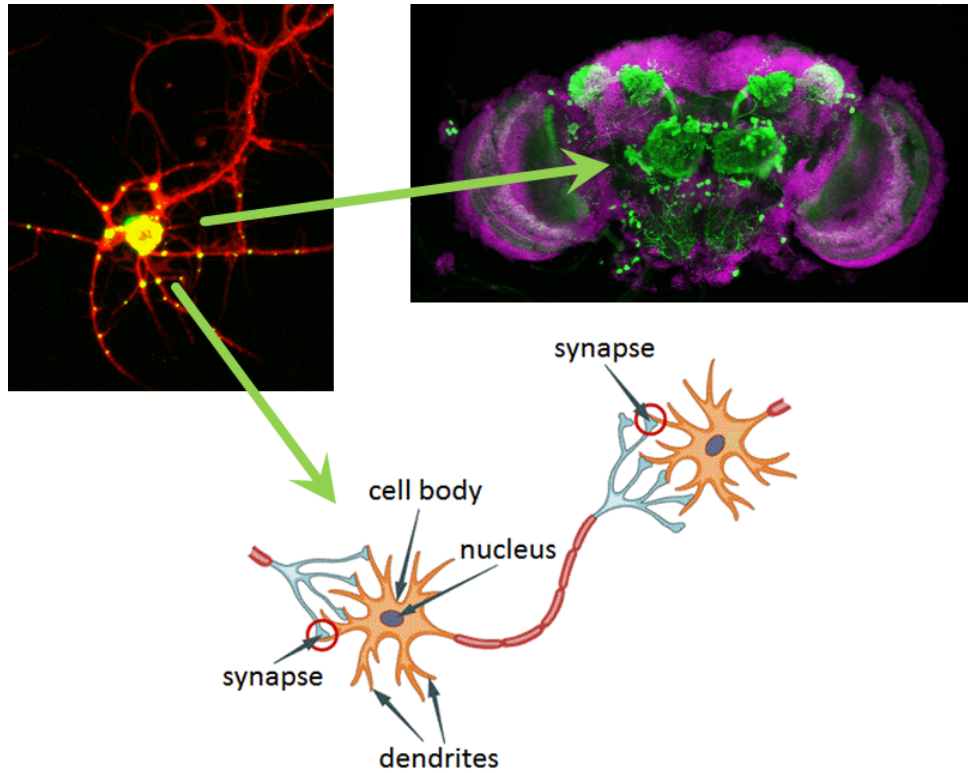
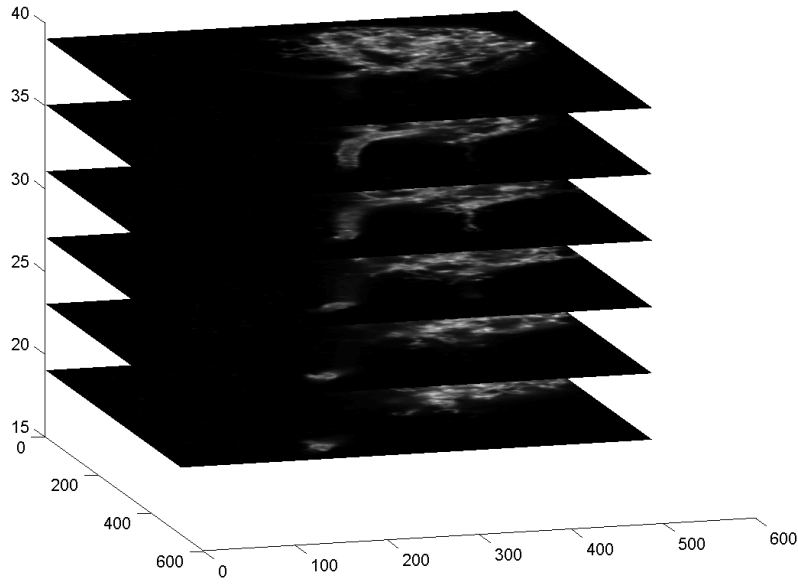


Figure 5.1: Illustration of neuron cell

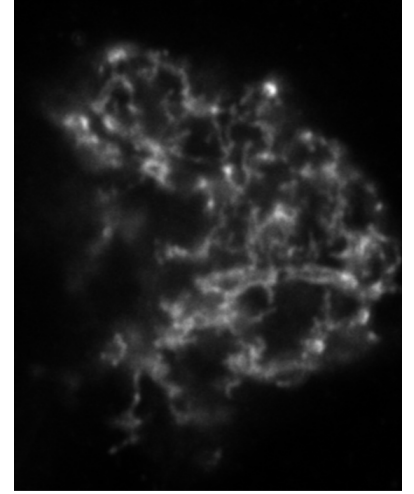
5.1 Introduction

As mentioned in *Chapter 3*, the biological study of memory formation focuses on investigating the roles of the different proteins in the process of the underlying protein synthesis. In *Chapter 3* and *4*, a shape modelling algorithm (Spotnick) is proposed in order to help the biologists to analyse the protein particles observed from living specimens. Spotnick detects the protein particles from the observed images, and performs colocalization analysis of the detected particles of different types of proteins.

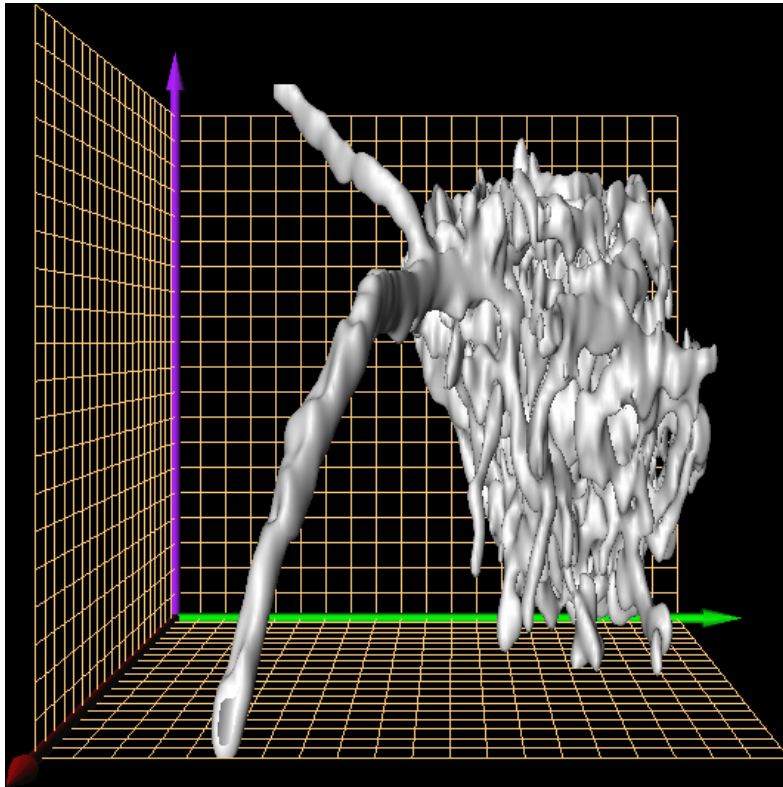
The shape information of the synapses where the protein synthesis takes place and where the particles are located is also an important aspect of the investigation. To observe the synaptic objects from the living specimens (see Fig. 5.1), the synapses are labelled by staining them with fluorescent markers, and the fluorescent images of the synapses are scanned by a Zeiss LSM 510 confocal microscope. Since the synapses are 3D objects, the shape information of each synapse is captured by scanning the synaptic object at multiple different ‘depths’ from top to bottom of the same specimen, and the shape information of the synapse is stored in the stack of the serially scanned images (slices). The goal of image processing in this work is then to segment the 3D shape of the synapse from a given image stack (as shown in Fig. 5.2(a)). The two main challenges in this work are the irregular luminance of synapses and the presence of out-of-focus features.



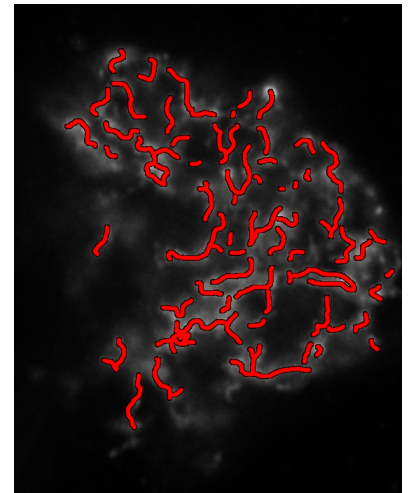
(a)



(b)



(c)



(d)

Figure 5.2: Sample of image stack and the 3D segmentation. (a) a stack of 2D synapse image slices captured at different depth of a neuron specimen. (b) a portion of a slice in (a). (c) a 3D segmentation of (a). (d) the manual segmentation of the in-focus ‘branches’ of the synaptic object.

5.1.1 Challenges in Synapse Segmentation

As illustrated in Fig. 5.1, the synapses are the ‘branch’ objects at the far end of the neuron cells, and connect to the dendrites of the other cells for transferring electrical signals. The protein synthesis for memory formation takes place at the synapses where the neuron cells are touched. The actual shape of the synapses are even more complicated, and a real image of the synapses (a portion of a slice in the image stack in Fig. 5.2(a)) is shown in Fig. 5.2(b). Each actual synapse has multiple small winding ‘branches’ which are shown as the ‘ridges’ in the image. Since the synapses are distributed in 3D, the complete shapes of the synaptic ‘branches’ may not be able to be illustrated by a single image but instead need to combine the ‘ridges’ of multiple slices.

However, the ‘ridges’ have varied intensities in the each slice, which means the ‘branches’ of the synapses have different brightnesses in the image slice. This is due to the defect of the staining process. When a synapse is stained with the fluorescent markers, the markers are usually distributed unevenly over the object. The branches of the synapse stained with different densities of markers generate different intensity of the light signal to the detector of the microscope during the scanning process, and appear with irregular brightness in the observed image slices. As described in *Section 2.2.1*, confocal microscopy can eliminate out-of-focus signals when detecting the light produced by fluorescence on the focal plane, but this confocal effect is controlled by the pinhole size in the aperture of the microscope. For the synapses, in order to capture the weak light signal from the weakly staining branches without over-exposing the ‘strong’ branches, the pinhole size needs to be enlarged during the observation. However, the larger pinhole size increases the ‘out-of-focus’ blur in the resultant images.

The ‘out-of-focus’ blur widens the actual widths of the in-focus branches (observed at the focal plane) in each resultant image. Also, the blur along the ‘z-axis’ (the axis perpendicular to the focal plane) gives rise to the ‘out-of-focus’ features in the images. To clarify, when the in-focus branches are scanned by the microscope at a certain depth of the specimen, the ‘background branches’ located slightly above or below that depth (focal plane) are also projected in the resultant image as ‘out-of-focus’ features. Moreover, a synaptic branch that is imaged as in-focus at one slice can be also scanned as out-of-focus at the other slices depending on the brightness of the branch. The goal of this work is to overcome this ‘focusing’ problem and segment the in-focus features with approximately actual widths from the given image stack. The segmentation is to label the in-focus features as the ‘object’, and label the rest of the stack (includes the out-of-focus features and the background) as the ‘background’.

5.1.2 Scope of the Segmentation Model

Shape fitting may be a solution for solving the two problems by modelling the shapes of the in-focus branches, but due to the high variation and complexity in the synapse shape and the non-uniform widths of the synaptic branches, this approach becomes impractical in this case. Ideally, the in-focus features can be distinguished from the out-of-focus features based on their

sharper and brighter appearance in the image. However, due to the co-existence of the non-uniform synapse luminance, the segmentation in the each image slice can not simply rely on the brightness as some out-of-focus features (from the strong branches) can be brighter than the in-focus features of the ‘weak’ branches. Moreover, since the branches are distributed in 3-D, the actual brightness of a branch can be affected by the background branches where a region with denser markers has higher local background luminance. This can cause missed detections in the segmentation in the bright regions when the dimmer features may be classified as ‘background’.

Instead, the ‘sharpness’ is a key for the segmentation. As shown in Fig. 5.2(b) and the manual segmentation (ground-truth) in Fig. 5.2(d), the ‘object’ features tend to have sharper appearance than the ‘background’ features (both the out-of-focus features and the ‘blur’ adjacent to the in-focus features). In fact, the ‘weak’ (dimmer) branches that are selected by the biological experts are based on the sharpness of the branches. On the other hand, ‘strong’ (bright) branches that are surrounded by the ‘blur’ in the bright regions are selected also based on the sharpness of the features in those regions. This means the ‘sharpness’ is a possible solution for segmenting the potential ‘object’ features. More importantly, the ‘sharpness’ is approximately invariant to the luminance or the shape of the object, so that the potential ‘object’ features can be detected regardless of the ‘strength’ (the brightness) of the branches.

In signal processing, the ‘sharpness’ is considered as the relative high frequency content, and a popular application for localizing signals both in space and scale/frequency in image processing is the wavelet transform (WT). By converting each image slice into the wavelet domain, the ‘sharpness’ of the features in the image can be represented by the wavelet features at each pixel site that consist of the high frequency wavelet content. Suppose the collection of wavelet features forms separate ‘clusters’ for the potential ‘object’ and the ‘background’ in a wavelet feature space, a given feature can then be classified as the ‘object’ or ‘background’ by ‘locating’ its wavelet feature in the ‘clusters’. Based on this assumption, and since the WT is independent of each slice, the wavelet approach is applied as an intra-slice (within each slice) segmentation. Since the wavelet feature is approximately invariant to the luminance, some background noise may have similar wavelet features to the ‘object’ and hence causes false alarms. In such cases, the global intensity model of the image stack can be used to prevent the false alarms as the background noise has significantly lower brightness than the synapse features.

However, the classification of the actual in-focus features can not be simply based on the intra-slice approaches but also need to consider the characteristics of the features across the slices (inter-slice) when the slices are observed at different depths in the same specimen. Since a branch is in-focus when it is observed at the focal plane (at a particular depth) but becomes out-of-focus at the other slices away from its focal plane, the in-focus projection(s) of the branch can be found by comparing all of the projections of the branch at different slices. Although brightness can not be used to identify in-focus objects when performing intra-slice segmentation, it can apply to the ‘vertical’ intensities of the same branch through the slices because the marker density

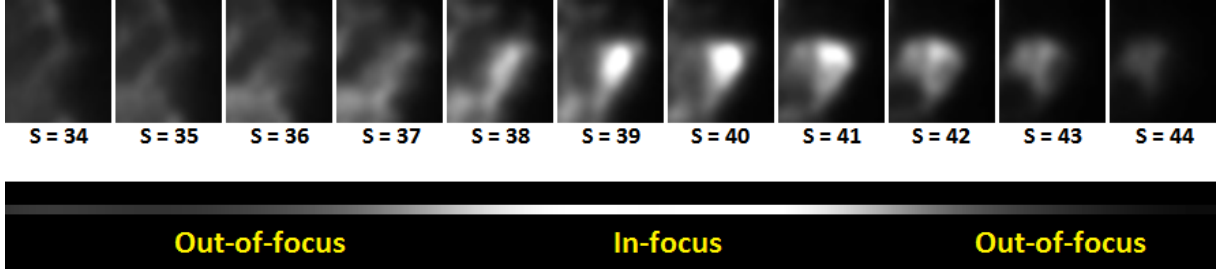


Figure 5.3: ‘Focusing’ condition in the inter-slice brightness

remains unchanged in each branch. As shown in Fig. 5.3, the brightness of a branch increases from being ‘out-of-focus’ to being ‘in-focus’ when ‘moving down’ vertically through the slices. Also, brightness decreases when moving deeper from the ‘in-focus’ slice to the ‘out-of-focus’ slices. In other words, by exposing the image stack ‘vertically’ along the z-axis, each synaptic feature has the highest intensity at the in-focus slice, and becomes ‘background’ in the other slices when the intensity is decreased below a particular threshold. Accordingly, the inter-slice intensity can help to locate the in-focus versions of the features.

Consequently, the intra-slice WT approach segments the potential ‘object’ features with irregular brightness from each slice and the ‘inter-slice’ intensity approach filters out the ‘out-of-focus’ features. In this work, a Bayesian framework is used to integrate these constraints. The idea is to consider the overall probability of a given feature being ‘object’/‘background’ based on the likelihoods derived from the wavelet features, the global intensity histogram, and the local ‘vertical’ intensity profile respectively. The wavelet-likelihood is the probability of the wavelet feature being in the ‘object’/‘background’ cluster. The global intensity likelihood is the probability of the intensity values being the potential ‘object’ or the background noise. The vertical intensity likelihood is the probability in the density distribution for the vertical intensity. Moreover, since the ‘branches’ are connected, a smoothness prior is introduced in order to encourage the segmentation of long thin structures. This prior enhances the likelihoods to form a Bayesian inference model for the segmentation in this work.

In the following sections, the Bayesian framework is firstly briefly reviewed in *Section 5.2*. Subsequently, the two likelihood terms are described in *Section 5.3* and the prior is presented in *Section 5.4*. Finally, some segmentation results of the introduced model are shown in *Section 5.6*.

5.2 Bayesian Segmentation Model

In the Bayesian worldview, the optimal value (i.e. the class or label) of some unknown field $\vec{i}$ can be found by maximizing the posterior probability density function (PDF) of $\vec{i}$ given a set of observed data Y . The maximum-a-posterior (MAP) estimate of $\vec{i}$ can be found by factorizing

the posterior using Bayes law as follows

$$p(\vec{i}|Y) = \frac{p(Y|\vec{i})p(\vec{i})}{p(Y)} \quad (5.1)$$

where $p(Y|\vec{i})$ and $p(\vec{i})$ are commonly referred to as the data likelihood and the prior respectively. The likelihood term specifies the probability model of the observed data given a value of the unknown field $\vec{i}$, while the prior term represents the PDF of $\vec{i}$ before any data is observed. As $p(Y)$ is constant with respect to $\vec{i}$, its calculation is usually omitted.

For 3D synapse segmentation, the segmentation problem becomes a classification problem that labels the pixels as ‘object’ or ‘background’ in each image slice. Given the intensity information ‘ $\mathbf{I}$ ’ of an image stack with M slices, the goal of the segmentation model is to assign labels ‘ B ’ to each slice and maximize the probability that is defined according to the Bayes’ rule as follows

$$\begin{aligned} \max_{B_1, \dots, B_M} p(B_1, \dots, B_M | \mathbf{I}_1, \dots, \mathbf{I}_M) &= \max_{B_1, \dots, B_M} \left(\frac{p(\mathbf{I}_1, \dots, \mathbf{I}_M | B_1, \dots, B_M) p(B_1, \dots, B_M)}{p(\mathbf{I}_1, \dots, \mathbf{I}_M)} \right) \\ &= \max_{B_1, \dots, B_M} (p(\mathbf{I}_1, \dots, \mathbf{I}_M | B_1, \dots, B_M) p(B_1, \dots, B_M)) \end{aligned} \quad (5.2)$$

where $\mathbf{I}_s$ is the intensity matrix of image slice s and B_s is the corresponding label field for this slice. Note that, the label $(b_s(\vec{x}))$ for a pixel $(\vec{x})$ at slice s has the value

$$b_s(\vec{x}) = \beta = \begin{cases} 1 & \text{for } \vec{x} \text{ in 'object'} \\ 0 & \text{for } \vec{x} \text{ in 'background'} \end{cases}$$

In order to make the optimization problem more tractable, each individual slice is optimized separately by assuming the labels of other slices are already known. Thus, the goal is to maximize the posterior for each slice defined as follows

$$p(B_s | \mathbf{I}_1, \dots, \mathbf{I}_M, B_1, \dots, B_M) \propto p_L(\mathbf{I}_1, \dots, \mathbf{I}_M | B_s) p_S(B_s | B_1, \dots, B_M) \quad (5.3)$$

where $p_L(\cdot)$ and $p_S(\cdot)$ are respectively the likelihood and prior of the model.

Furthermore, the assumption is made that the probabilities for each pixel in a given slice are independent. Therefore, the posterior is written as

$$p(B_s | \mathbf{I}_1, \dots, \mathbf{I}_M, B_1, \dots, B_{s-1}(\vec{x}), B_{s+1}(\vec{x}), \dots, B_M) \propto \prod_{\vec{x} \in X} p_L(I_1(\vec{x}), \dots, I_M(\vec{x}) | b_s(\vec{x})) p_S(b_s(\vec{x}) | \mathcal{B}(-\vec{x})) \quad (5.4)$$

when X is the set of the pixels in slice s . Note that $\mathcal{B}(-\vec{x})$ is the segmentation state of all the pixels in the image stack except for the current pixel $\vec{x}$. Usually, the posterior is further simplified by modelling it as a Markov Random Field (MRF). The idea of MRF is to assign a neighbourhood window N_{3D} for each considered pixel $\vec{x}$, where the probability of $\vec{x}$ is only

dependent on the neighbouring pixels inside N_{3D} but not the pixels outside the window. By using MRF, the posterior in Eq. (5.4) can be simplified as

$$p(B_s | \mathbf{I}_1, \dots, \mathbf{I}_M, B_1, \dots, B_{s-1}(\vec{x}), B_{s+1}(\vec{x}), \dots, B_M) \propto \prod_{\vec{x} \in X} p_L(I_1(\vec{x}), \dots, I_M(\vec{x}) | b_s(\vec{x})) p_S(b_s(\vec{x}) | B_{N_{3D}}(\vec{x})) \quad (5.5)$$

where $B_{N_{3D}}(\vec{x})$ represents the labels of the neighbour pixels in N_{3D} . The likelihood term $p_L(\cdot)$ consists of the intra-slice likelihood and the inter-slice likelihood which are derived respectively based on the wavelet features and the local vertical intensity. The prior $p_S(\cdot)$ focuses on the connection of the synaptic features and the smoothness of the overall object in 3D.

5.3 Likelihoods

The likelihood term $p_L(\cdot)$ in Eq. (5.4) is factorized further in terms of three separate likelihoods as follows

$$\begin{aligned} p_L(I_1(\vec{x}), \dots, I_M(\vec{x}) | b_s(\vec{x})) &\propto (p_W(\mathbf{W}_s(\vec{x}) | b_s(\vec{x})))^{\alpha_W(\beta)} \\ &\times (p_G(I_s(\vec{x}) | b_s(\vec{x})))^{\alpha_G(\beta)} \\ &\times (p_V(I_1(\vec{x}), \dots, I_s(\vec{x}), \dots, I_M(\vec{x}) | b_s(\vec{x})))^{\alpha_V(\beta)} \end{aligned} \quad (5.6)$$

The first likelihood is the wavelet-likelihood $p_W(\cdot)$ designed to classify the image features with synaptic ‘texture’ as ‘object’ regardless of luminance of the features in the given slice. The second one is the global intensity likelihood $p_G(\cdot)$ and is intended to discriminate against regions of low-intensity in order to remove the false alarms caused by noise features with similar texture to the synaptic objects. Since these two likelihoods only consider the image information in each individual slice, they form the intra-slice part of $p_L(\cdot)$. Conversely, the third likelihood forms the inter-slice part by considering the local intensities across the slices. The vertical intensity likelihood $p_V(\cdot)$ is designed to detect in-focus features and is derived from the vertical intensity profile at a given location. Since these likelihood represent different image characteristics of the object, the effects are controlled by the weights $\{\alpha_W(\beta), \alpha_G(\beta), \alpha_V(\beta)\}$ respectively.

5.3.1 Wavelet-likelihood

Since the target synaptic object has irregular intensity in the image stack, a simple intensity-based likelihood alone would be infeasible for the segmentation. Instead, a preferable way is to classify pixels based on the luminance invariant texture content. The wavelet transform is ideally suited for this task since the discrete wavelet transform (DWT) decomposes an image into a luminance dependent low frequency band and a number of luminance invariant high frequency sub-bands. The idea is to associate a ‘wavelet feature’ to each pixel by grouping the high frequency content extracted from the sub-bands in the wavelet domain excluding the low frequency band. An assumption is made in this work that the ‘object’ wavelet features are well separated from the ‘background’ wavelet features at each scale in the wavelet domain, which

means the clusters of ‘object’ wavelet features are distant from the clusters of ‘background’ wavelet features. Hence the probability of a given pixel belonging to either of the classes can be measured by how close that it is located toward to the corresponding clusters. In other words, the probability distribution for the wavelet-likelihood $p_w(\cdot)$ is equivalent to the distribution density of the corresponding feature space.

Therefore, the ‘wavelet feature’ and the cluster density for each class are the keys for measuring $p_w(\cdot)$. In this work, a wavelet feature is defined as a vector where the entries are the wavelet coefficients from the high frequency sub-bands, when a slice is represented by the wavelet coefficients in different sub-bands after the WT. The cluster density for ‘object’/‘background’ is measured from the distribution formed by the wavelet features of manually labelled ‘object’/‘background’ regions in the training data (as shown in Fig. 5.2(d)). Each cluster density is parameterized by fitting Gaussian mixture models (GMMs) to the corresponding distribution. The resultant GMMs are then the PDFs for measuring $p_w(\cdot)$ for pixels in each slice. Clearly, the fundamental step of the wavelet-likelihood estimation is the form of the WT used to calculate the wavelet coefficients for each slice.

The discrete wavelet transform (DWT) is a well known technique in image compression [2,47] and has become popular in the texture synthesis [81], image de-noising [65] and watermarking [62]. The idea of the DWT is to convert a signal into the wavelet domain by convolving it with two particular 1-D wavelet kernels which behave like a low pass filter (G) and a high pass filter (H) respectively. For images, as shown in Fig. 5.4, the convolution with 2-D wavelet kernels (the low pass filter and the high-pass filters along different orientation) can be simplified by convolving the image with 1-D kernels firstly along the row axis and then along the column axis. As a result, the low frequency band (lo-band: ‘LL’) is a down-sampled version of the original image, and the high frequency sub-bands (hi-bands: ‘LH’, ‘HL’ and ‘HH’) contains the wavelet coefficients (the magnitudes of the frequencies) for same wavelet but at different orientations ($0^\circ, 90^\circ, \pm 45^\circ$) respectively. Consequently, the image features can then be represented by the wavelet coefficients with high values. Also, the DWT can be a multi-scale process where the WT is iterated on the ‘LL’ band in the higher scale and that further divides the low frequency band into low and high sub-bands, and hence the band-pass wavelet coefficients of the image feature cover more of the frequency range of the signal.

5.3.1.1 Dual-tree Complex Wavelet Transform (DTCWT)

Although the DWT is unlike the discrete Fourier transform (DFT) in that it can capture both frequency and location information of the image features, as discussed by Kingsbury *et al.* in [87], it suffers from the defect of shift variance. This defect happens when an input image is displaced slightly, the wavelet features do not appear to displace similarity due to aliasing in the lower resolution bands. This means that similar image features that are located differently in the image can have very different wavelet features in the DWT domain. In order to generate

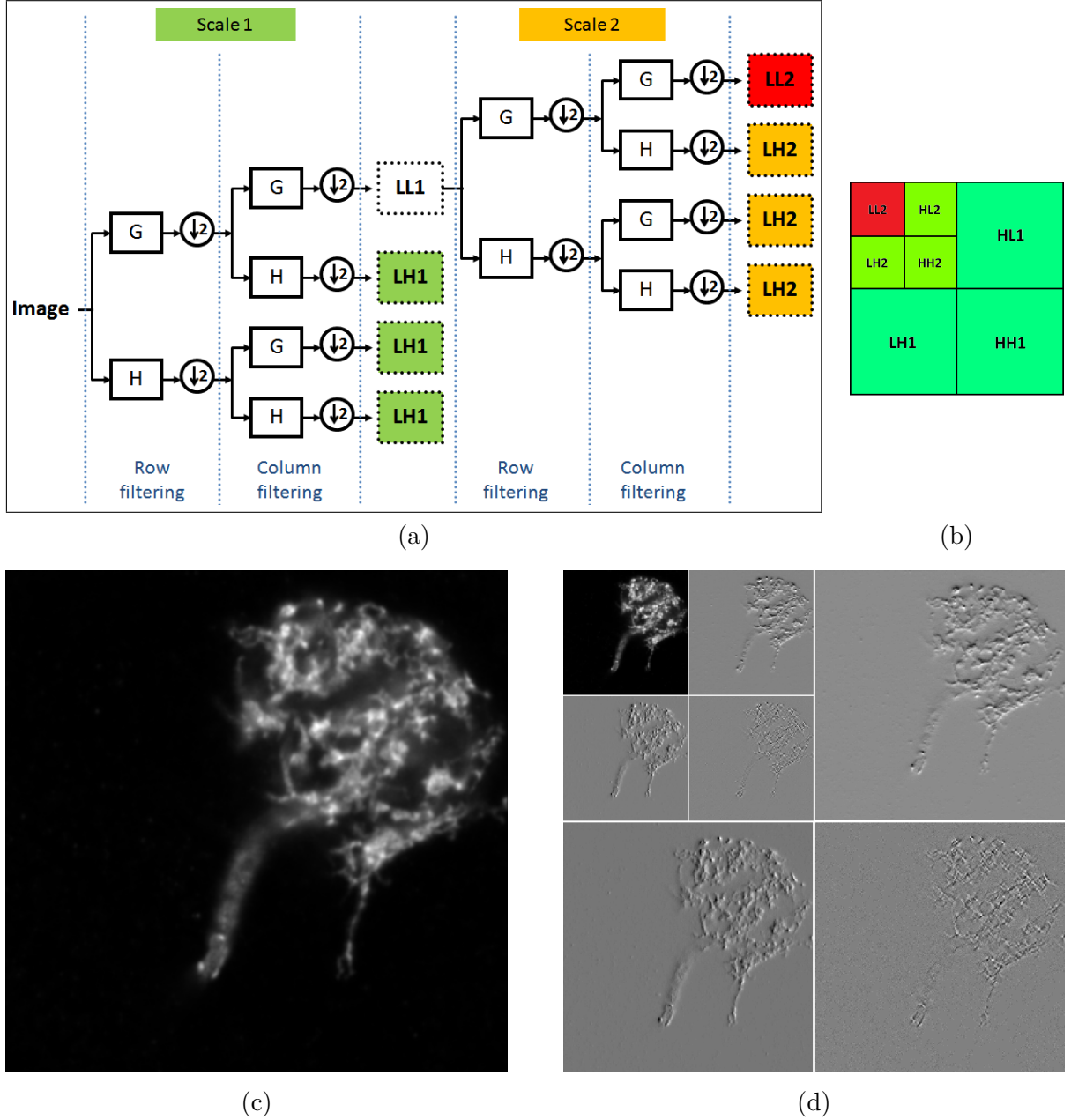


Figure 5.4: Illustration and sample of DWT. (a) An illustration of the process of multiscale 2D DWT with 2 scales where G and H are the ‘low-pass’ and ‘high-pass’ wavelet kernels respectively. (b) the usual structure and the size relationship of the different sub-bands after the process in (a). (c) A sample image for the DWT. (d) the sub-band wavelet coefficients of (c) distributed in the structure of (b).

‘regularized’ wavelet features for the image features, a WT that can eliminate the shift variance is preferable. In [87], Kingsbury *et al.* introduced the dual-tree complex wavelet transform (DTCWT) to resolve the shift variance defect in the WT, and the keys of this technique are the ‘complex wavelet’ and the ‘dual-tree’ structure. The DTCWT was motivated by the fact that the shift variance does not happen in the case of DFT, and the reason is that the DFT is based on complex-valued oscillating sinusoids that are analytic basis functions, but the DWT is based on real-valued oscillating wavelets. Hence, by carefully designing complex wavelets such that the real and imaginary components of the wavelet are 90° out of phase with each other, the shift variance can be mitigated in the WT. The ‘dual-tree’ structure is one of the approaches based on the idea of complex wavelets.

As shown in Fig. 5.5(a), in DTCWT, two wavelet trees are computed, one for the real coefficients and the other for the imaginary coefficients. This means that the DTCWT produces complex wavelet coefficients without involving any complex-valued filters in the estimation process. Consequently, the construction of the complex wavelets is transferred to the design of set of real-valued 1-D wavelet kernels (i.e. G_0 and G_1 , H_0 and H_1 , where G_0 and H_0 are ‘low-pass’, G_1 and H_1 are ‘high-pass’) that can be coupled as Hilbert transform pairs (i.e 90° out of phase). The detail of designing the wavelet kernels can be found in [87]. The application of the ‘complex wavelet’ can regularize the wavelet responses for similar image features and separate the wavelet responses along two different directions in each hi-band into two individual hi-bands.

Like the DWT, the DTCWT is a multi-scale transform, where each scale of the transformed image consists of one lo-band and the six hi-bands (‘LHa’, ‘LHb’, ‘HLa’, ‘HLb’, ‘HHa’ and ‘HHb’). The hi-bands correspond to the response of the image to the ‘complex wavelet’ at six different orientations ($\pm 15^\circ$, $\pm 45^\circ$ and $\pm 75^\circ$) in a frequency band dictated by the scale of the wavelet. The hi-band wavelet coefficients have complex values that indicate both the magnitude and phase responses of the wavelet at the corresponding orientation. Subsequently, the DTCWT is applied instead of the traditional DWT to abstract the shift-invariant wavelet features.

5.3.1.2 Wavelet Features and the Gaussian Mixture Model

When each slice in the given stack is transformed to the DTCWT domain, each pixel in the slice is associated with a wavelet feature at each scale of the transformation. In this work, a multiscale DTCWT with 3 scales (a 2-scale DTCWT is illustrated in Fig. 5.5) is performed, and hence each pixel is associated with a set of three wavelet features, where one wavelet feature

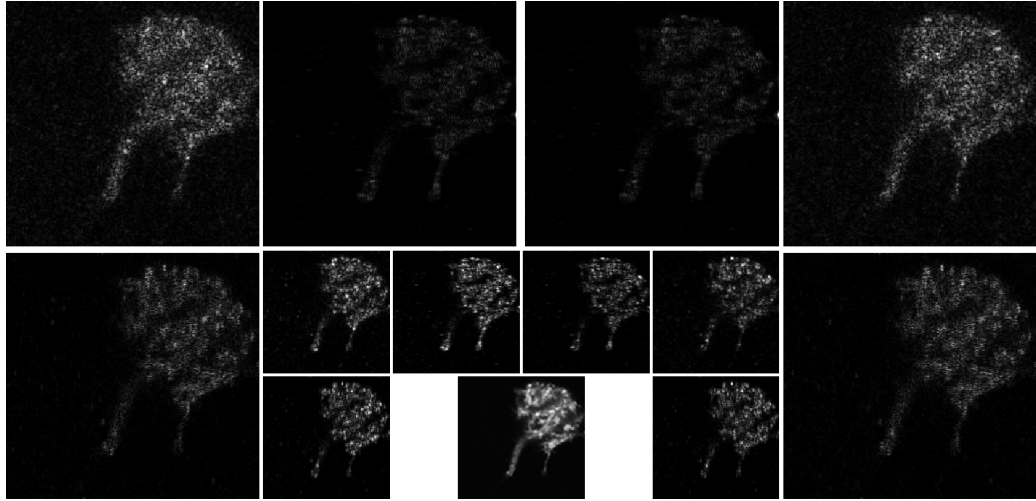
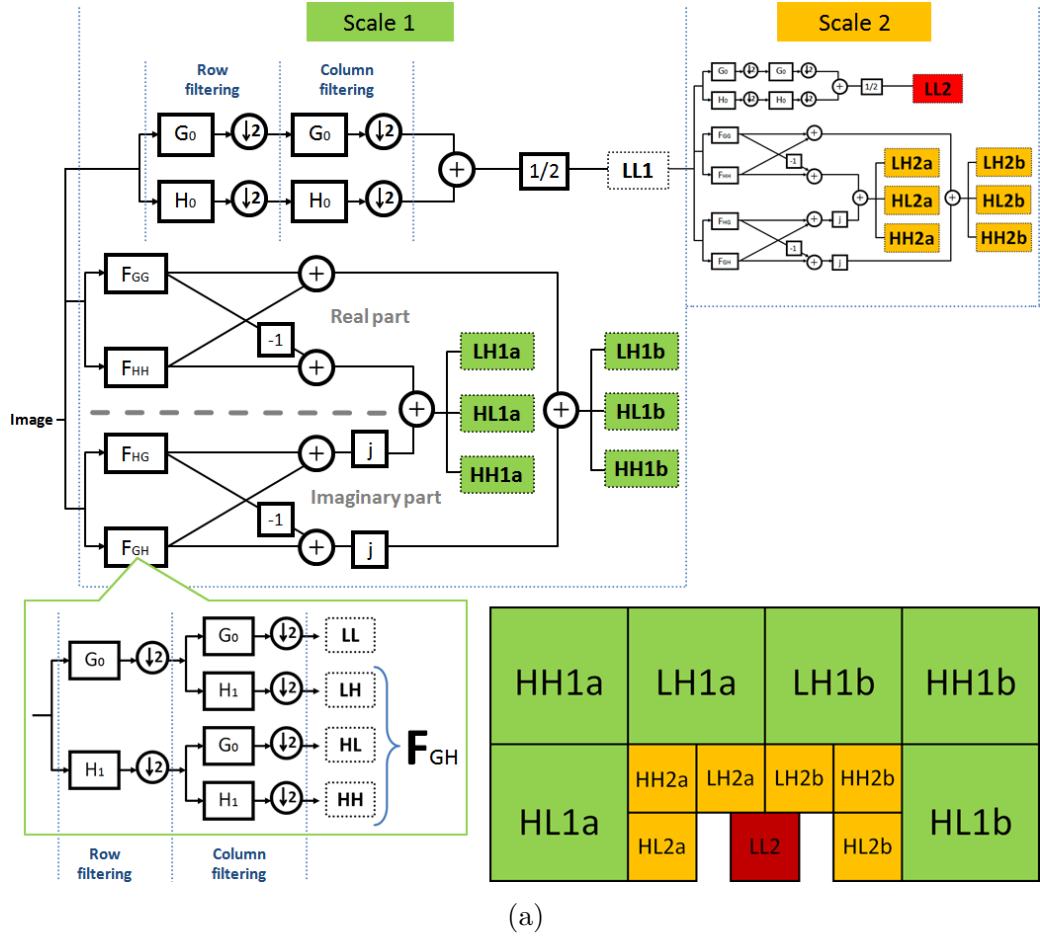


Figure 5.5: Illustration and sample of DTCWT. (a) An illustration of the process of multiscale 2D DTCWT with 2 scales. Each of $\{F_{GG}, F_{HH}, F_{HG}, F_{GH}\}$ is a 2D filtering process where F_{GH} is a process with ‘low-pass’ wavelet kernel G_0 and ‘high-pass’ wavelet kernel H_1 . (b) the sub-band wavelet coefficients for Fig. 5.4(c) after the 2-scale DTCWT.

from each scale. Given a pixel $\vec{x}$ in the slice s , the wavelet feature at scale h is a 12-D vector

$$W_{s,h}(\vec{x}) = \begin{bmatrix} \text{Re} \left\{ \begin{matrix} w_{HLa}(\vec{x}) \\ w_{HLb}(\vec{x}) \\ w_{LHa}(\vec{x}) \\ w_{LHb}(\vec{x}) \\ w_{HHa}(\vec{x}) \\ w_{HHb}(\vec{x}) \end{matrix} \right\} \\ \text{Im} \left\{ \begin{matrix} w_{HLa}(\vec{x}) \\ w_{HLb}(\vec{x}) \\ w_{LHa}(\vec{x}) \\ w_{LHb}(\vec{x}) \\ w_{HHa}(\vec{x}) \\ w_{HHb}(\vec{x}) \end{matrix} \right\} \end{bmatrix}$$

where the entries are real parts ($Re\{\cdot\}$) and the imaginary parts ($Im\{\cdot\}$) of the corresponding wavelet coefficients $w(\cdot)$ in the 6 hi-bands (see Fig. 5.5(b)). In addition, the pixels are assigned wavelet features at lower resolution scales using the quad-tree down-sampling relationship.

Since each pixel has one $W_{s,h}(\cdot)$ for each scale of the DTCWT, the ‘object’/‘background’ regions also have one cluster density for $W_{s,h}(\cdot)$ at each scale. In order to estimate the cluster densities, a set of segmented training data for the ‘object’/‘background’ regions is manually selected from some sample images by the ground-truth evaluation of biological experts (as shown in Fig. 5.2(d)). Note that segmented training data does not cover all the image features but only the features that are confidently classified as ‘object’. For each segmented image slice, a binary mask of the ‘object’ pixels and the background pixels is provided. After the DTCWT, the binary masks are used to filter the wavelet features for the labelled pixels, and generate a set of training wavelet features for each class at each scale. Each set of the training data is subject to the 12-D wavelet feature space, and the distribution of the data points forms the density. In order to estimate the density function, a GMM is fitted to the distribution. This approach is the traditional application of GMM for data clustering as described in *Chapter 3*. The initial parameter estimation for the GMM is first generated from the training data using the mean-shift clustering algorithm [27]. The GMM parameters are then refined by applying the expectation maximization (EM) algorithm for data clustering to the initial estimation.

After the training process, there are two different GMMs for wavelet features at each scale of the WT, and $\Theta_{h,\beta}$ is the parameter set for the ‘object’/‘background’ region GMMs at scale h . Each GMM is a PDF for estimating the wavelet-likelihood $p_w(\cdot)$ of a given pixel $\vec{x}$ at slice s being ‘object’/‘background’. Since there are multiple GMMs (from different scales) for each class, the wavelet-likelihood is defined by the product of the GMMs of all scales for the same

class. Therefore, $p_w(\cdot)$ for pixel $\vec{x}$ with wavelet feature set $\mathbf{W}_s(\vec{x})$

$$p_w(\mathbf{W}_s(\vec{x})|b_s(\vec{x}) = \beta) = \prod_{h=1}^3 G(W_{s,h}(\vec{x})|\Theta(h, \beta)) \quad \beta \in [0, 1] \quad (5.7)$$

where H is the number of scales of the DTCWT being used in the model, $G(\cdot)$ is the GMM with the parameter set $\Theta(\cdot)$ for label β at scale h .

The result of maximum likelihood segmentation based on the wavelet likelihood only is shown in Fig. 5.6(c) of the given slice in Fig. 5.6(a). The wavelet-likelihood can detect the synaptic features, however it also detect some false alarms due to the background noise in the image, which is shown in the bottom left corner in Fig. 5.6(c). This noise is very dim and hence difficult to observe in the original image. To illustrate the noise more clearly, a ‘noise enhanced’ version of the image is shown in Fig. 5.6(b). Since the wavelet features are luminance invariant, the false alarm can occur in the inter-slice segmentation when the noise has similar wavelet features to the ‘object’ features even though they are dim. Hence, a global intensity likelihood is applied to assist the wavelet-likelihood as a means of eliminating the low intensity background noise (the result of combining the wavelet-likelihood and the global intensity likelihood is shown in Fig. 5.6(d)).

5.3.2 Global Intensity Likelihood

Fig. 5.6(a) and (b) show that the intensity of the background noise is very close to the background brightness and is significantly different from the synaptic features even for the weakly illuminated branches. Although the intensity is not suitable for distinguishing the in-focus and the out-of-focus features within each slice, it can be applied to eliminate the background noise (certainly are ‘background’ region), and hence allow the luminance-invariant wavelet approach to distinguish the synaptic features (possible ‘objects’ region) without being affected by the noise. Based on observation, as shown in Fig. 5.7, the background cluster and the potential ‘object’ cluster (represented by the peaks) are separable from each other in the intensity histogram. The idea of the global intensity likelihood $p_G(\cdot)$ is to estimate the probability of a given pixel being ‘object’/‘background’ based on how close the intensity is toward to the corresponding mode in the histogram.

Intuitively, $p_G(\cdot)$ for each class can be estimated by the density of the corresponding cluster, and this requires the density estimations. Due to the bell shape appearance of the two modes, the ‘object’ and ‘background’ intensity can be fitted by two different Gaussian-PDFs. Consequently, the intensity histogram of the given image stack is fitted with a GMM with two components, and the model parameters are estimated via the EM algorithm. The peaks are the initial means and the initial mixing weights are set by the ratio of the peaks’ values. To initialize the covariances, the training data mentioned in *Section 5.3.1.2* is separated into two pixel sets, the set of the ‘object’ pixels and the set of the remaining pixels. The initial covariances are computed by fitting a Gaussian to the intensity distribution of each pixel set. After refining the

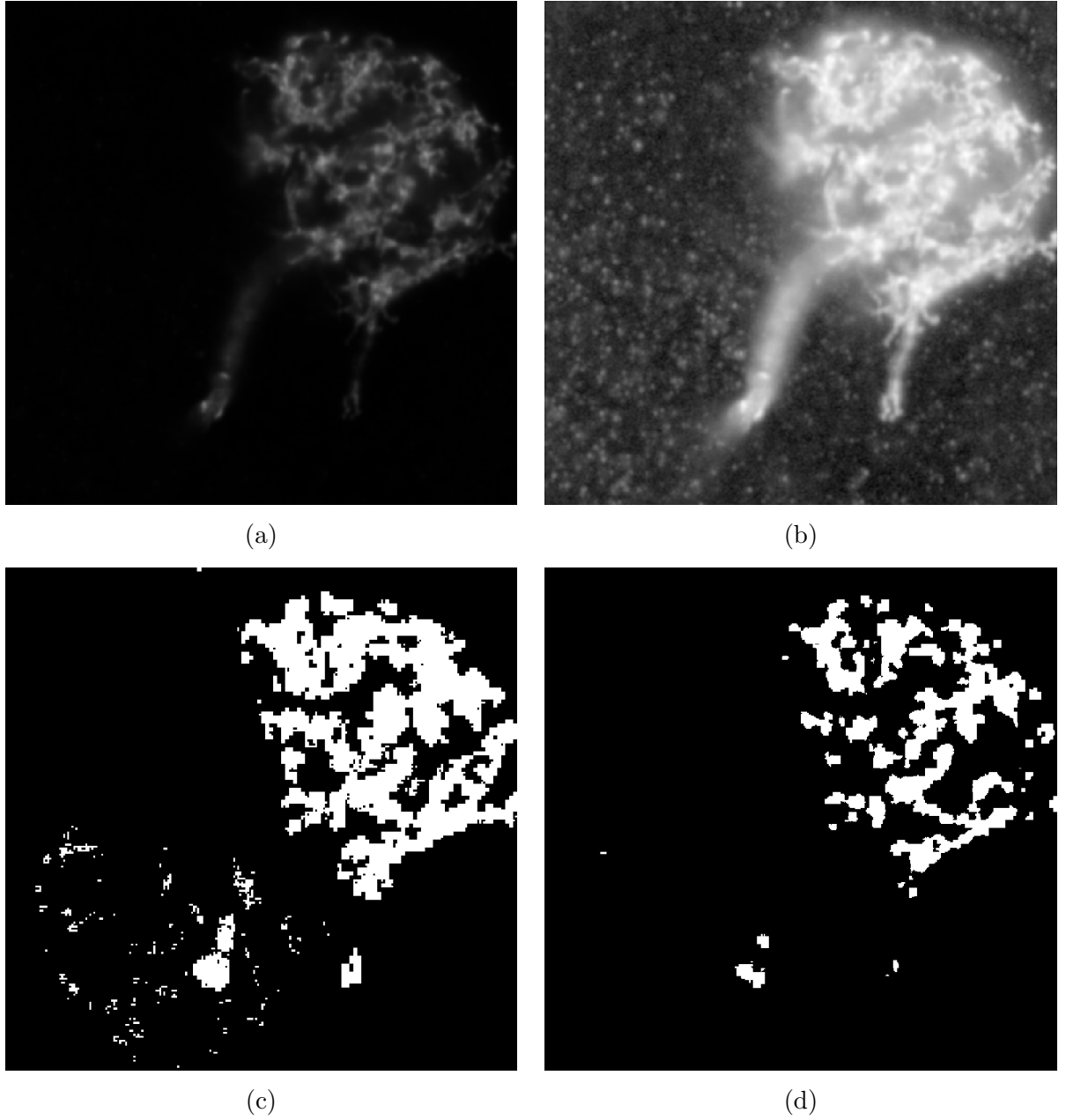


Figure 5.6: Maximum Likelihood segmentation based on the intra-slice likelihoods. (a) is an image at slice $s = 30$ of the stack in Fig. 5.2. (b) is ‘noise enhanced’ version of (a) using a gamma operation. (c) is the segmentation result obtained by Maximum Likelihood estimation base only on the wavelet-likelihood $p_W(\cdot)$. (d) is the segmentation result obtained by Maximum Likelihood estimation based on both $p_W(\cdot)$ and the global intensity likelihood $p_G(\cdot)$. The false detections in (c) due to noise are corrected in (d) by incorporating $p_G(\cdot)$.

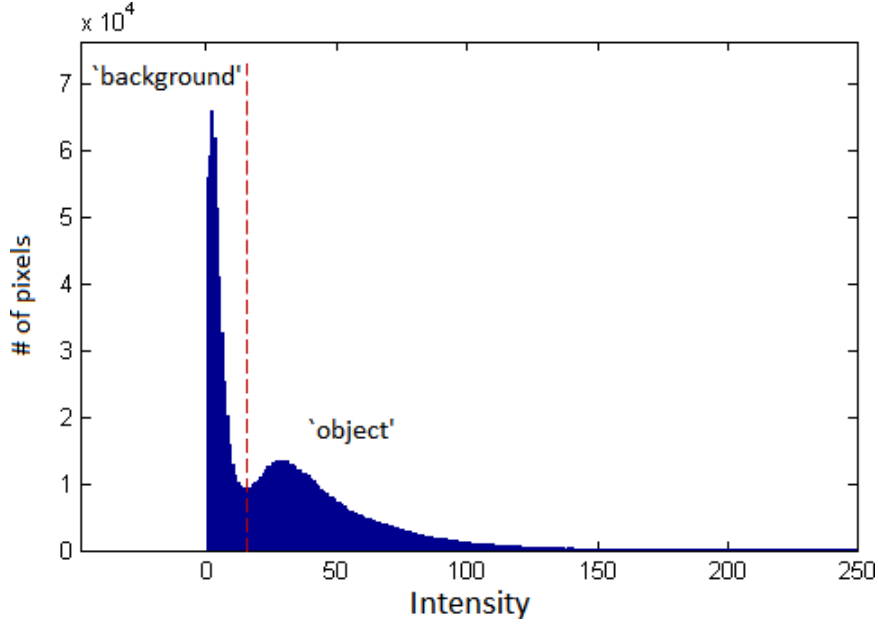


Figure 5.7: Histogram of global intensity of the image stack. *There are two peaks in the histogram where the peak with a mean at approximate 40 belongs to ‘object’ and the other narrower peak belongs to ‘background’. We model both peaks with Gaussian-PDFs.*

initial estimation with the EM process, the resultant means and covariances are the parameters $\theta_G(\beta) = \{\mu_G(\beta), \sigma_G^2(\beta)\}$ of the Gaussian-PDFs for the intensity clusters.

The density functions of $p_G(\cdot)$ are then set by the Gaussian-PDFs estimate, and hence the likelihood probability $p_G(\cdot)$ of a given pixel being ‘object’ or ‘background’ as follows

$$p_G(I_s(\vec{x})|b_s(\vec{x}) = \beta) = \begin{cases} \exp\left(-\frac{(I_s(\vec{x}) - \mu_G(0))^2}{2\sigma_G^2(0)}\right) & , \beta = 0 \\ \exp\left(-\frac{(I_s(\vec{x}) - \mu_G(1))^2}{2\sigma_G^2(1)}\right) & , \beta = 1 \end{cases} \quad (5.8)$$

The result of the segmentation model with only $p_W(\cdot)$ and $p_G(\cdot)$ is given in Fig. 5.6(d), and obviously, the background noise features are removed while comparing to Fig. 5.6(c) which is based on $p_W(\cdot)$ only.

The combination of $p_W(\cdot)$ and $p_G(\cdot)$ forms the intra-slice likelihood in Eq. (5.6). This intra-slice likelihood overcomes the irregular brightness problem by detecting potential ‘object’ features using wavelet feature that are invariant to luminance, and at same time reject the low intensity noise textures by using the global intensity histogram (as shown in Fig. 5.6(d)). In order to further filter out the ‘out-of-focus’ features, the last likelihood $p_V(\cdot)$ is applied to incorporate with the intra-slice likelihood using the inter-slice intensity profile.

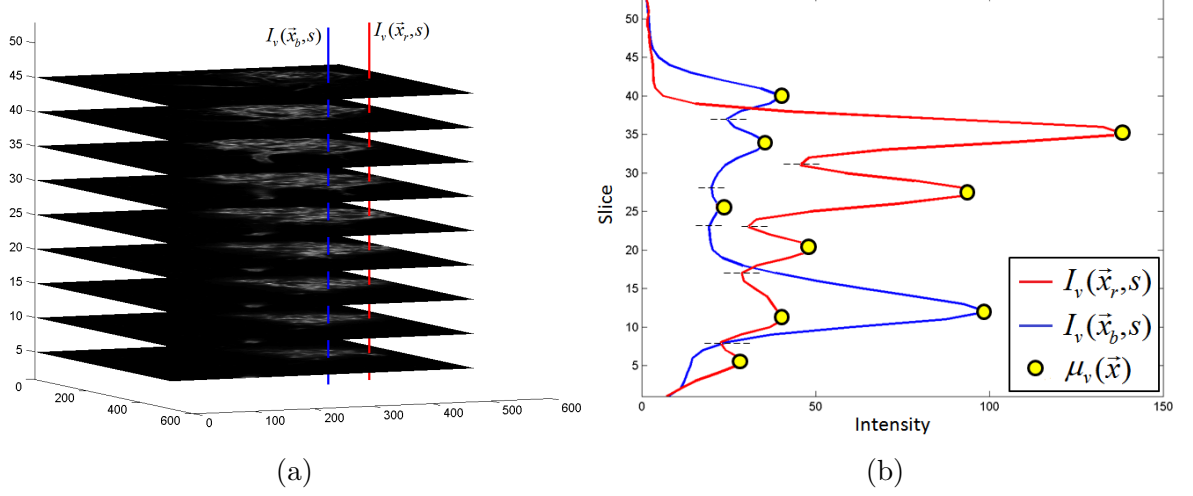


Figure 5.8: Local vertical intensity profiles. (a) is the image stack in Fig. 5.2 but highlighted by two vertical inter-slice lines at different pixel-wise locations. (b) shows the vertical intensity profiles corresponding to the locations labelled in (a)

5.3.3 Inter-slice Intensity Likelihood

As discussed in Section 5.1, the inter-slice intensity profile of a feature through the slices can help to determine where the features are in-focus when the intensity is relatively high. Two examples of the inter-slice intensity profile (the ‘red’ and ‘blue’ plots in Fig. 5.8(b)) are given in Fig. 5.8. Each intensity profile is a 1-D intensity distribution ($I_v(\vec{x}, s)$) along the same pixel coordinate ($\vec{x}$) through the slices (s) in the given image stack.

Each peak in the profile corresponds to the projection of the same synaptic structure at different slices in the stack. For a given peak, some slices correspond to when the synaptic object is in-focus and other slices correspond to out-of-focus projections (as shown in Fig. 5.3). Since the synaptic branches are arranged in 3D, each inter-slice intensity profile of the stack can have multiple peaks in the distribution. Hence, a peak in the profile $I_v(\vec{x}, S)$ that has the mode at slice s is the local intensity profile of the actual feature located at coordinate $\vec{x}$ and at depth s . The mode slice s of the peak is approximately (due to the step size on the z-axis during the scanning process) the focal plane for the feature, and is where the feature brightness is highest. As the scanning ‘moves’ toward to s , the feature that is projected comes into focus, and the intensity in the projection increases. When the scanning reaches s , the feature is certainly in focus and has the brightest projection. As the scanning passes s and starts to move away from s , the feature becomes out-of-focus again in the projection and the brightness decreases.

The problem is to decide where the feature becomes out-of-focus as the intensity decreases from the maximum of the local inter-slice intensity profile, and this is concerned by the inter-slice likelihood $p_v(\cdot)$. The idea of $p_v(\cdot)$ is to measure the probability of a feature projection belonging

to the ‘object’ regions when given the difference between the intensity in the projection and the maximum of the local inter-slice profile of the feature. This means $p_V(\cdot)$ is based on the distribution of the inter-slice intensity of the feature. Since $p_V(\cdot)$ is estimated based on the intensity difference between the considering projection and the highest intensity in the peak, the model for $p_V(\cdot)$ could be assumed to be a Gaussian-PDF.

However, as shown in Fig. 5.8(b), the ‘spread’ of the local profiles are not uniform. This is due to the variation in the widths of the synaptic branches and the orientation of the branches. The branches that are thicker occupy more slices, and hence have more in-focus projections and have wider inter-slice intensity profiles. Conversely, the thinner branches occupy fewer slices (or just one) and have narrower profiles. Similarly, when a branch is not distributed along the scanning plane but is orientated toward the z-axis, the branch can occupy multiple slices and hence have a wider inter-slice profile. On the other hand, the different magnitudes of the peaks are due to the irregular luminance of the branches.

Since all objects in the same image stack are captured by the microscope under the same setting, the point-spread-function (PSF), i.e. the cause of the out-of-focus blur, is the same for the objects in the same image stack. Also, the PSF is assumed as Gaussian in 3D [98]. Accordingly, if a feature exists only on a single horizontal slice, the local inter-slice intensity profile will have a Gaussian shape with a fixed variance (or roll-off rate). Apparent variations on the roll-off rate (as shown in Fig. 5.8(b)) indicate that the object is in-focus on a number of sequential slices of the same position (e.g. it may be oriented along the axis perpendicular to the plane of the slices).

Consequently, an assumption is made that the focus state of the feature projections is not based on the roll-off rate or the magnitudes of the local inter-slice intensity profile but is more related on the intensity difference with respect to the maximum of the local inter-slice profile of the feature. Hence, instead of using the distributions of different peaks as the likelihood model for the corresponding features, a Gaussian-PDF with fixed variance σ_v^2 is applied as the model of $p_V(\cdot)$. The mean of the Gaussian-PDF is a variable $\mu_v(\vec{x})$ depending on feature at $\vec{x}$. Intuitively, it is set by the maximum of the local inter-slice profile (i.e. the ‘yellow’ dots in Fig. 5.8(b)) of the feature being considered. When there are multiple peaks in the inter-slice intensity profiles of the image stack (the ‘red’ and ‘blue’ plots in Fig. 5.8(b)), the peaks are separated by the local minima (highlighted by the black dash lines in Fig. 5.8(b)) in the profiles. The feature projections that are within the same peak belong to the same actual feature and share the same $\mu_v(\cdot)$ in the $p_V(\cdot)$ estimation. Furthermore, based on observation, the value of σ_v is set to 8.5 in this work.

On the other hand, $p_V(\cdot)$ for the ‘background’ is set to 1. The reason of this approach is that the intra-slice likelihood will classify both the in-focus and out-of-focus features as ‘object’ (as shown in Fig. 5.6(d)). Hence, $p_V(\cdot)$ for the ‘object’ can be considered as the weights that reduce the possibly of the out-of-focus features being selected as ‘object’. Therefore, the inter-slice intensity likelihood $p_V(\cdot)$ is defined as follows

$$p_V(I_1(\vec{x}), I_s(\vec{x}), \dots, I_M(\vec{x}) | b_s(\vec{x}) = \beta) \propto \begin{cases} 1 & , \beta = 0 \\ \frac{1}{\sigma_v \sqrt{2\pi}} \exp\left(-\frac{(I_s(\vec{x}) - \mu_v(\vec{x}))^2}{2\sigma_v^2}\right) & , \beta = 1 \end{cases} \quad (5.9)$$

The combination of the three likelihoods can enable the model to segment the synaptic feature with varied brightness, and filters out the out-of-focus features. In fact, a single synapse is an object with connected ‘branches’ in the specimen, and therefore the ‘object’ features should also be connected as the ‘branches’ in the image stack. However, the likelihood terms do not include any a priori knowledge of the smoothness properties of the object features in the image stack. For the cases that some synaptic ‘branches’ are stained with little fluorescent material in the specimen, the ‘weak’ appearance of those ‘branches’ in the image stack makes it likely to be classified as ‘background’ by the likelihood terms of the model. As a result, the object of a single synapse in the image stack may be ‘broken’ into several disconnected objects after segmentation. In order to enhance the connection of such ‘weak’ branches, a spatial smoothness prior that derived from the connectivity (the ‘frame’) of the synapses is injected to the segmentation mode. The detail of this prior is described in the next section.

5.4 Smoothness Frame Prior

This idea of the smoothness frame prior $p_s(\cdot)$ is to label a pixel as ‘object’ or ‘background’ based on the label of the neighbouring pixels. But different to the traditional smoothness prior, the neighbourhood smoothness in this work is constrained by the connection of the synaptic branches. This connection is referred to as the ‘frame’ $\mathbf{B}_F$ of the synapses which can be considered as the frame of the connected ‘branches’ in the image stack, and hence the prior term is called the ‘frame prior’. The frame prior is designed to augment the standard isotropic smoothness prior by boosting smoothness along the axis of the synapse frame while penalizing smoothness perpendicular to the frame.

The frame constraint on the prior is represented by using two different sets of neighbouring pixels $\vec{y}$ at slice s_y within the 3D neighbourhood window $\mathbf{N}_{3D}$ during the prior estimation. The two sets of the pixels are the neighbour pixels inside the frame (in-frame, i.e. $B_F(\vec{y}, s_y) = 1$), and the neighbour pixels outside the frame (out-frame, i.e. $B_F(\vec{y}, s_y) = 0$). The neighbour pixels in each set are weighted using the same Gaussian smoothness weights ($\omega_F(\cdot)$), but the combination of the two sets are controlled by a constant factor λ . For $\beta = 1$, the in-frame set is weighted by λ and the out-frame set is weighted by $(1 - \lambda)$. Conversely, for $\beta = 0$, the in-frame and out-frame sets are weighted by $(1 - \lambda)$ and λ respectively. The reason of this configuration is to avoid over-biasing the segmentation $\mathbf{B}$ of image stack toward the frame $\mathbf{B}_F$. The ‘strength’ of ‘frame constrained’ part is controlled by the ratio $\lambda \in (0, 1)$ in order to vary the ‘strength’ of the connections between the synapses during the segmentation. Therefore, the frame prior $p_s(\cdot)$

in Eq. (5.5) is expressed as

$$p_S(b_s(\vec{x})|\cdot) \propto \begin{cases} \lambda \sum_{(\vec{y}, s_y) \in \mathbf{N}_{3D}} \left(\omega_i |1 - B_F(\vec{y}, s_y)| |1 - b_{s_y}(\vec{y})| \right) + (1 - \lambda) \sum_{(\vec{y}, s_y) \in \mathbf{N}_{3D}} \left(\omega_i |B_F(\vec{y}, s_y)| |1 - b_{s_y}(\vec{y})| \right) & \beta=0 \\ \lambda \sum_{(\vec{y}, s_y) \in \mathbf{N}_{3D}} \left(\omega_i |B_F(\vec{y}, s_y)| |b_{s_y}(\vec{y})| \right) + (1 - \lambda) \sum_{(\vec{y}, s_y) \in \mathbf{N}_{3D}} \left(\omega_i |1 - B_F(\vec{y}, s_y)| |1 - b_{s_y}(\vec{y})| \right) & \beta=1 \end{cases} \quad (5.10)$$

where $\mathbf{N}_{3D}$ is a spherical neighbourhood region with radius $r = 2$. $\omega_i = \omega_F(\vec{y}, s_y)$ is the Gaussian weight associated to the neighbour pixel where a close neighbour has high value of ω_i , and $\sum \omega_i = 1$.

To obtain the ‘frame’ of the synaptic object, a method (as shown in Fig. 5.9) similar to the Laplacian pyramid algorithm introduced by Burt and Adelson in [20] is applied to the given image stack. The Laplacian pyramid is a decomposition of the original image into a hierarchy of images such that each level corresponds to a different band of image frequencies. The Laplacian pyramid is a useful tool in image processing and computer vision [33, 37, 44, 56, 61], as it provides an overcomplete signal representation, and can be treated as a frame operator [45]. In [45], the idea of frame operator is to highlight the ‘frame’ (high frequency sub-band) of the sharp features in the given images by subtracting blurred versions of the images from the original images (a Laplacian process).

When the Laplacian process is performed on a given slice, the resultant ‘frame’ image would contain both positive and negative values. Since the coarse image is a blurred version of the original image, the regions inside the synaptic objects (bright features) would tend to have positive values in the ‘frame’ image, and the dim regions in the original image tend to be negative. Intuitively, the frame of the synapses can be extracted by binary-masking the positive features in the ‘frame’ image. The Laplacian stack algorithm is a multi-scale process that repeats the Laplacian operation between the coarse image and a coarser version. The difference between every two adjacent scales is placed as a Laplacian stack, where the finest ‘frame’ image is on the top and the coarsest version is located at the bottom. Similar to the wavelet transform, the Laplacian stack algorithm is a simple approach to divide the high frequency contents in each sub-band of the image by repetitively subtracting the lower sub-band (the coarser version) from the low frequency band (the coarse version). This means that the ‘frame’ images (high frequency contents) depend on the sharpness rather than the luminance of the image objects and hence the resultant frame has approximately regular features for the branches with varied brightness.

As shown in Fig. 5.9, the Laplacian stack model applied in this work has two modifications to the algorithm in [45]. The first modification extends the original algorithm to 3-D in order to extract the synaptic frame distribution in 3-D. This means the model is performed on the whole image stack rather than each slice. The coarse stacks at different scales are generated by filtering the adjacent finer stacks with the same 3-D Gaussian kernel. In practice, the 3-D convolution is

separable by filtering the stack with a 1-D Gaussian kernel (with covariance $\sigma^2 = 1$) firstly along the row then the column and finally along the intra-slice axis. A Laplacian approach is applied to the image stack pyramid to get the 1st order Laplacian features by subtracting a coarse image stack from the adjacent finner image stack above it, and as a result, a stack of the 1st order Laplacian features at different scale is produced.

The second modification is the additional 2nd-order Laplacian (L^2) that applied to the original 1st-order Laplacian (L^1) pyramid, and extracting the synaptic frame from the L^2 pyramid. The reason of this modification is to extract a frame with thin and connected branches. The goal of the synaptic frame is to enforce the connection of the features with affecting the width of the branches, but wider frames would cover the adjacent ‘blur’ to the actual in-focus features and hence possibly cause false alarm in the segmentation. The frames extracted from the finer versions in the L^1 stack are slender, however some connections are broken. On the other hand, the results from coarser versions (Fig. 5.10(c)) have strong connectivity but the widths of the branches are too wide which may cause false alarm in the segmentation, especially for the out-of-focus features. In order to find the frame but remain the connectivity, a Laplacian process is performed on the L^1 pyramid to thin the Laplacian features by narrowing the bandwidth of the frequency contents. This generates the L^2 pyramid and the frame extracted from this pyramid is the same by thresholding the positive L^2 features. A comparison between the L^1 frame and the L^2 frame is shown in Fig. 5.10.

Based on observation, the frame extracted from scale 6 of the L^2 pyramid provides thin and connected frame of the synapses and hence is regarded as the frame of the synapses. The ‘likely’ connections between the synaptic branches are identified by the frame $\mathbf{B}_F$ and is subjected to the prior estimation in Eq. (5.10).

5.5 Graph-cuts Optimization

After setting up the likelihoods and the frame prior, the final step of the segmentation model is to optimize the Bayesian framework. In this work, the optimization is accomplished using the Graph-cuts algorithm, and hence the segmentation model is named Synapcut. Graph-cut was firstly applied by Greig *et al.* to maximize a posteriori estimation in binary segmentation in [42]. As discussed in the review in *Chapter 2*, graph-cut is a popular segmentation method for efficiently solving the problems formulated in terms of energy minimization by reducing the problems to the max-flow/min-cut optimization. The main advantage of the using Graph-cuts technique over the other optimization methods such as the iterated conditional modes (ICM) [9] is that it finds the global minimum of the defined energy function for a binary labelling problem.

The energy functions typically used in Graph-cuts are similar to those used in Bayesian frameworks using MRFs. The labelling energy function for Graph-cuts segmentation of in this work is expressed as

$$E(\mathbf{B}) = \mathbf{R}(\mathbf{B}) + \mathbf{C}(\mathbf{B}) \quad (5.11)$$

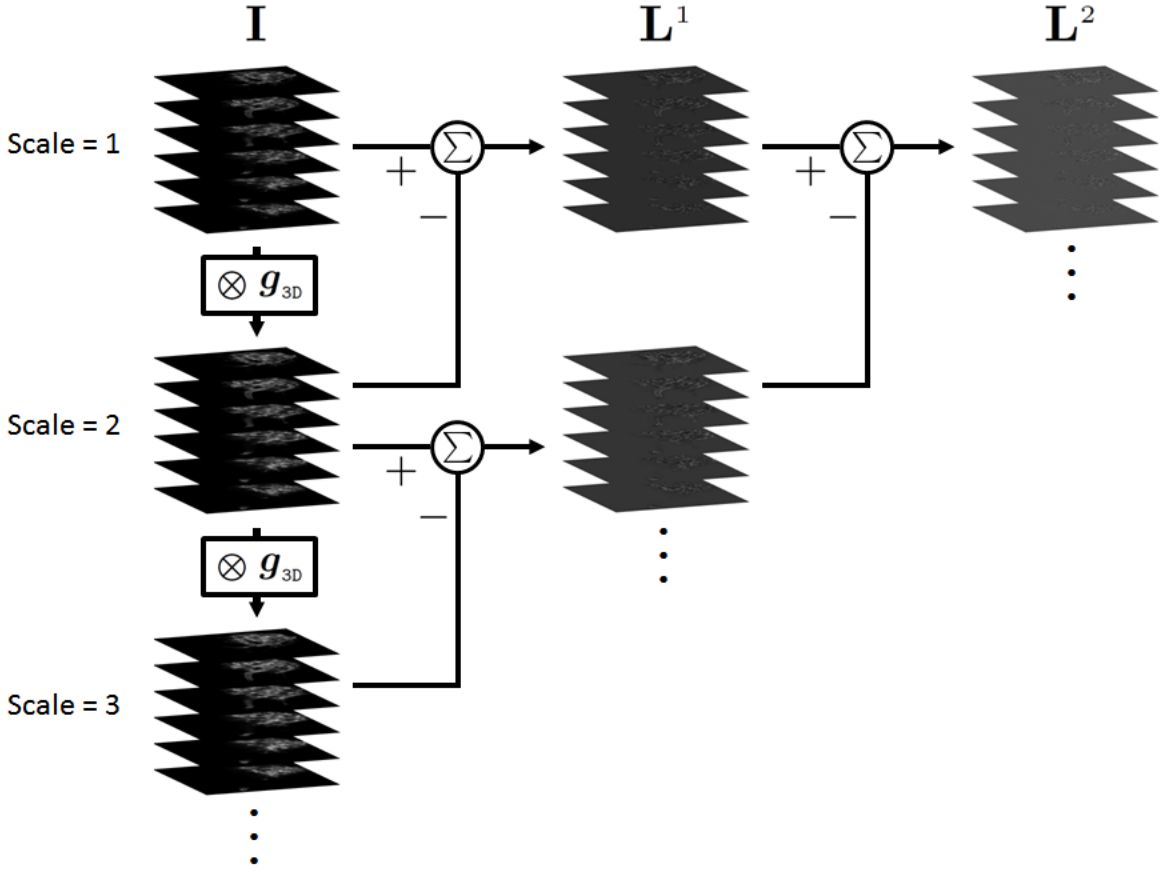


Figure 5.9: Illustrates the process of the Laplacian approach. The left column shows the image stack with different ‘resolutions’ by convolving with a 3D Gaussian kernel g_{3D} . The stacks are arranged from fine to coarse. The middle column contains the 1st order Laplacian features which obtained by abstracting the coarse image stack from the ‘finer’ one ‘above’ it. The right column consists of the 2nd order Laplacian features that obtained by subtracting the 1st order Laplacian features at different resolution levels.

where

$$\begin{aligned}
 R(B) &= \sum_{\tilde{\mathbf{x}} \in \mathbf{X}, \mathbf{s} \in \mathbf{S}} \mathbf{R}_{\mathbf{x}}(\mathbf{b}_{\mathbf{s}}(\tilde{\mathbf{x}})) \\
 C(B) &= \sum_{\tilde{\mathbf{x}} \in \mathbf{X}, \mathbf{s} \in \mathbf{S}} \sum_{(\tilde{\mathbf{y}}, \mathbf{s}_{\mathbf{y}}) \in \mathbf{N}_{3D}} \mathbf{C}_{\{(\tilde{\mathbf{x}}, \mathbf{s}), (\tilde{\mathbf{y}}, \mathbf{s}_{\mathbf{y}})\}}
 \end{aligned} \tag{5.12}$$

where $E(B)$ is the total energy cost of labelling the given image stack as B . $R(B)$ gives the likelihood (or regional) energy of the stack labels B , and $R_x(b_s(\vec{x}))$ is the region energy for each pixel in the stack and is the cost of assigning $b_s(\vec{x})$ to 0 or 1. $C(B)$ gives the boundary cost of B , and $C_{\{(\vec{x}, s), (\vec{y}, s_y)\}}$ is the connection between the pixel $(\vec{x}, s)$ and its neighbour pixel $(\vec{y}, s_y)$, this is also the cost of a discontinuity in the labelling between $b_s(\vec{x})$ and the $b_{s_y}(\vec{y})$.

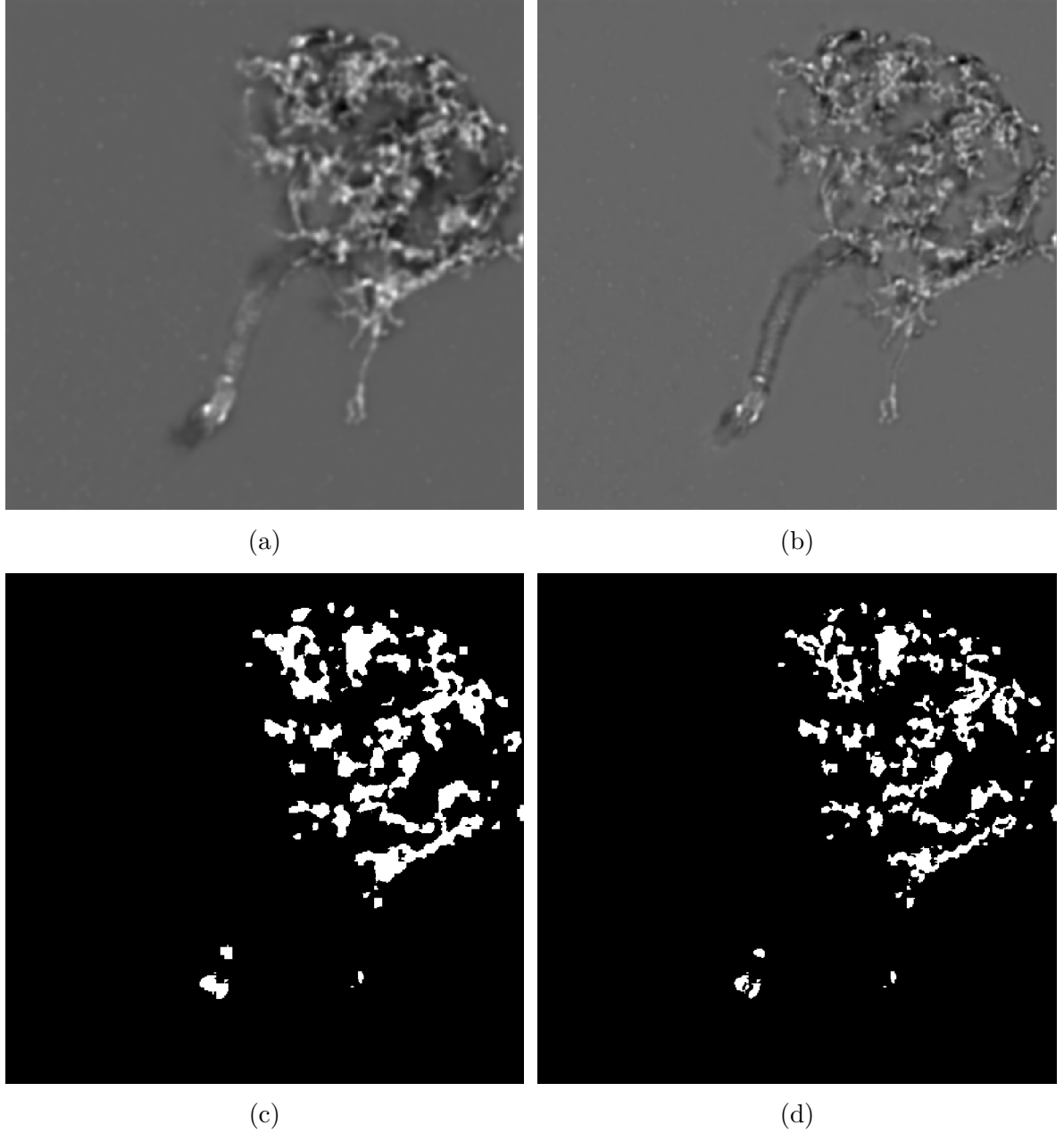


Figure 5.10: Illustration of the Laplacian process at slice $s = 30$ with respect to the image in Fig. 5.6(a). (a) shows the 1st order Laplacian features at scale 6 for the slice $s = 30$ which is shown in in Fig. 5.6(a). (b) shows the 2nd order Laplacian features derived from (a). (c) is the 1st order synapse frame by thresholding the positive 1st order Laplacian features in (a). (d) is the 2nd order synapse frame by thresholding the positive 2nd order Laplacian features in (b).

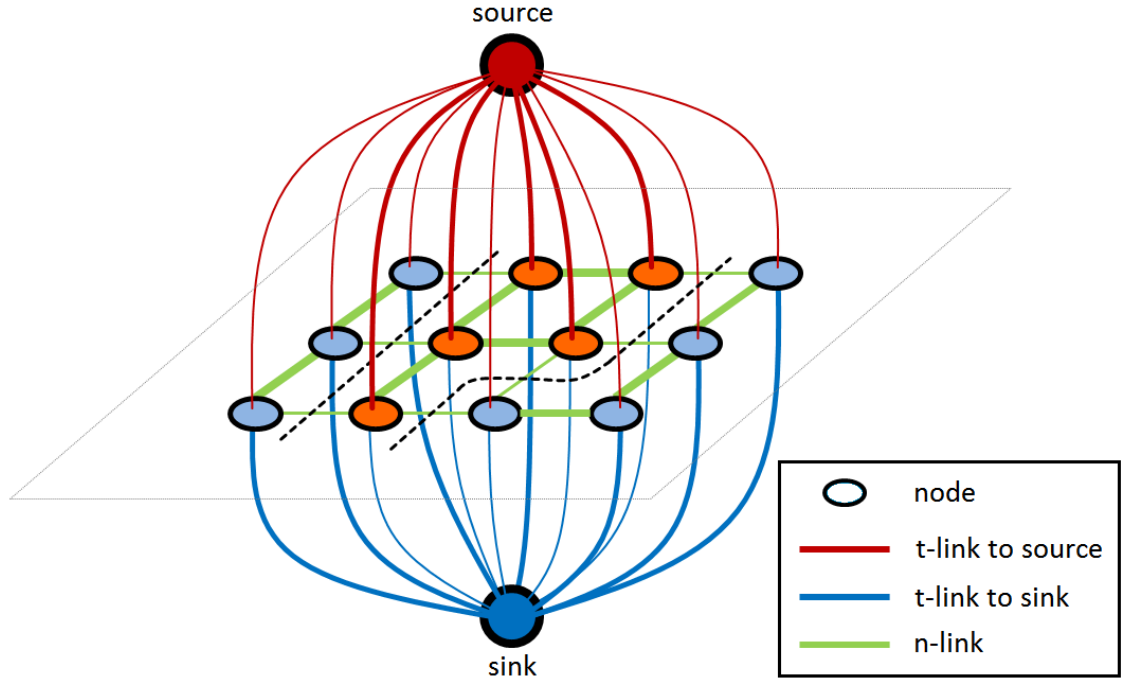


Figure 5.11: Graph-cut optimization

5.5.1 Graph-cut Model

For image segmentation using Graph-cut technique, the image is modelled as a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ which is a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. As shown in Fig. 5.11, each pixel $(\vec{x}, s)$, corresponds to a node which has an edge (the n-link) between itself and each of its neighbour $(\vec{y}, s_y) \in \mathbf{N}_{3D}$. Every node has also has two further links (the t-links) to a ‘source’ node and a ‘sink’ node which are associated with possible labels (0 or 1) for each node. The cost of each n-link is related to the boundary energy between the two pixels connected by the n-link, and the cost of the t-link is related to the energy of each pixel being assigned with either label.

The segmentation of the image is equivalent to cutting a subset of the edges in the graph such that the graph is divided into two separated sub-graphs, one consists of the source nodes and the other contains the sink nodes. The pixels belong to the source sub-graph are labelled as 1 (i.e. $b_s(\vec{x}) = 1$), and the pixels in the sink sub-graph are labelled as 0 (i.e. $b_s(\vec{x}) = 0$). The cost of the cut (the black dash lines in Fig. 5.11) is given by the cost of all the n-links and t-links which are cut to divide the graph, and hence the optimal segmentation for the image is given by the cut of minimum cost.

5.5.2 Graph-cut Segmentation

The minimum cut of the graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is equivalent to the minimization of the energy function in Eq. (5.11). The image stack segmentation can be solved by using the Graph-cut

technique when considering the whole stack as a graph model. The region energy $R(B)$ (t-link) for each pixel in the stack is derived from the combined likelihood of the pixel. On the other hand, the connections $C(B)$ (n-links) between each pixel and the neighbouring pixels are derived related the frame prior. In this case, the neighbouring structure $\mathbf{N}_{3D}$ for $C(B)$ is a 3D spherical region. This means $C(B)$ does not only consist of the connections between a pixel and each intra-slice neighbouring pixel but also includes the connections between the pixel and each inter-slice neighbouring pixel at other slice but within $\mathbf{N}_{3D}$.

Since the MAP estimation is mapped to the energy minimization, $R(B)$ are set by energy values converted from the combined likelihood probability $p_L(\cdot)$ in Eq. (5.6). The energy is estimated by the minus natural logarithm of the probability, and a large probability would give rise to a lower energy. However, if $R_x(b_s(\vec{x}) = 1)$ is set by the energy of $p_L(\cdot | b_s(\vec{x}) = 1)$, then a pixel with higher likelihood probability for belonging $\beta = 1$ would have lower energy to $\beta = 1$ is more likely to be cut and labelled as $\beta = 0$ after energy minimization. This does not match the MAP estimation. Conversely, if $R_x(b_s(\vec{x}) = 1)$ is set by the energy of $p_L(\cdot | b_s(\vec{x}) = 0)$, then a pixel with higher likelihood probability for belonging $\beta = 0$ is more likely to be labelled as 0 after the segmentation, and hence match the MAP estimation. Consequently, the region energy $R_x(b_s(\vec{x}))$ for the two labels are set as follows

$$\begin{aligned} R_x(b_s(\vec{x}) = 0) &= -\ln(p_L(\cdot | b_s(\vec{x}) = 1)) \\ R_x(b_s(\vec{x}) = 1) &= -\ln(p_L(\cdot | b_s(\vec{x}) = 0)) \end{aligned} \quad (5.13)$$

By substituting the individual likelihoods from Eq. (5.6), the equations in Eq. (5.13) become

$$\begin{aligned} R_x(b_s(\vec{x}) = 0) &= -\left(\alpha_W(1) \ln(p_W(\cdot | b_s(\vec{x}) = 1)) + \alpha_G(1) \ln(p_G(\cdot | b_s(\vec{x}) = 1)) + \alpha_V(1) \ln(p_V(\cdot | b_s(\vec{x}) = 1))\right) \\ R_x(b_s(\vec{x}) = 1) &= -\left(\alpha_W(0) \ln(p_W(\cdot | b_s(\vec{x}) = 0)) + \alpha_G(0) \ln(p_G(\cdot | b_s(\vec{x}) = 0)) + \alpha_V(0) \ln(p_V(\cdot | b_s(\vec{x}) = 0))\right) \end{aligned} \quad (5.14)$$

For the connection term $C(B)$, since $C_{\{(\vec{x},s),(\vec{y},s_y)\}}$ is the connection between the pixel $(\vec{x}, s)$ and its neighbour pixel $(\vec{y}, s_y) \in \mathbf{N}_{3D}$, it is intuitively set by the neighbouring weights that are applied for the frame prior estimation in Eq. (5.10). Consequently, $C_{\{(\vec{x},s),(\vec{y},s_y)\}}$ is defined as

$$C_{\{(\vec{x},s),(\vec{y},s_y)\}} = \begin{cases} \lambda\omega_i & , \text{ if } B_F(\vec{x}, s) = B_F(\vec{y}, s_y) \\ (1 - \lambda)\omega_i & , \text{ if } B_F(\vec{x}, s) \neq B_F(\vec{y}, s_y) \end{cases} \quad (5.15)$$

After setting up the t-links and the n-links of the graph of image stack, the max-flow/min-cut algorithm proposed by Boykov and Kolmogorov in [14] is used to find the segmentation. However, a direct Graph-cut process on the whole image stack would be extremely expensive and is impractical. An alternative approach is to estimate each image slice s separately by subjecting a sub-stack with 5 slices $\{s - 2, s - 1, s, s + 1, s + 2\}$ to the Graph-cut segmentation, and only the result for s is considered as final segmentation. The reason of chosen 5 slices for each sub-stack is because that $C(B)$ are estimated based the spherical neighbourhood region

$\mathbf{N}_{3D}$ with radius $r = 2$. The less number of slices would cause un-precise segmentation as some of the neighbouring connections are missing. Also, due to the same reason, only the middle slice of the result of each sub-stack is precisely segmented. The final results for all the slice that are estimated separately are combined to give the segmentation of the whole stack.

For the experiments in this work, the weights $\{\alpha_W(\cdot), \alpha_G(\cdot), \alpha_V(\cdot)\}$ in Eq. (5.6) are set as $\alpha_W(0) = 1$, $\alpha_W(1) = 3$, $\alpha_G(0) = 1$, $\alpha_G(1) = 1$, $\alpha_V(0) = 1$, $\alpha_V(1) = 6$ and λ is set to 0.8. The results for the real data are presented in the following section.

5.6 Segmentation Results

The 3D segmentation result of the given sample image stack is given in Fig. 5.2(b). As it is difficult to visualize a 3D segmentation, a number of selected small samples of the image stack and the segmentations for a number of slices in the stack are illustrated in Fig. 5.12-Fig. 5.15. Although this representation is not ideal for viewing the inter-slice relationships in the segmentation masks, it presents the effectiveness of aspects of the segmentation model. Also, the comparison between the estimation and the ground-truth in Fig. 5.2(d) is illustrated in Fig. 5.16.

As shown in the figures, Synapcut has successfully segmented the in-focus synaptic features in both bright regions dim regions. For example, in Fig. 5.13, the bright feature on the left of ‘slice 9’ is selected while the dimmer sharp in-focus features in ‘slice 15’ are also detected. The effectiveness of the vertical likelihood in detecting the in-focus regions can be seen in Fig. 5.14. As shown in centre of ‘slice 4’ to ‘slice 14’, when the object becomes brighter, there are more in-focus features being detected, but when it becomes dimmer in the later slices, there are less features being selected. However in some slices, like ‘slice 17’ in Fig. 5.14 and ‘slice 44’ in Fig. 5.15, the in-focus ‘constraint’ is too strict so that the ‘relatively’ in-focus features are unselected. The reason of this ‘false negative’ may be that the variance σ_v^2 for the inter-slice intensity likelihood $p_v(\cdot)$ is too small. Potential approaches to improve the $p_v(\cdot)$ estimation are to use a better variance, or to apply a different density model that is better fitted to the PSF of the microscope. Furthermore, the effectiveness of the frame prior is can seen in ‘slices 44 and 45’ in Fig. 5.12, the in-focus features with blob-like appearance would be detected as individual spots by only the likelihoods but are detected as elongated synaptic features.

The comparison with manual segmentation Fig. 5.16 shows that Synapcut segments the obvious synaptic features and also maintains the connectivity for the features. In Fig. 5.16, there are a number of connections indicated by the red-lines which are not detected by Synapcut. However, this does not necessarily indicate failure as in most cases Synapcut detects these regions in one of the neighbouring slices. Although the manual segmentation is made by the biologists, it is very subjective since there is no certain threshold value or criterion for distinguishing the in-focus and out-of-focus, and the decision is only based on personal judgement on the intra-slice ‘sharpness’ of the image features and the inter-slice intensity changes of the same feature. For

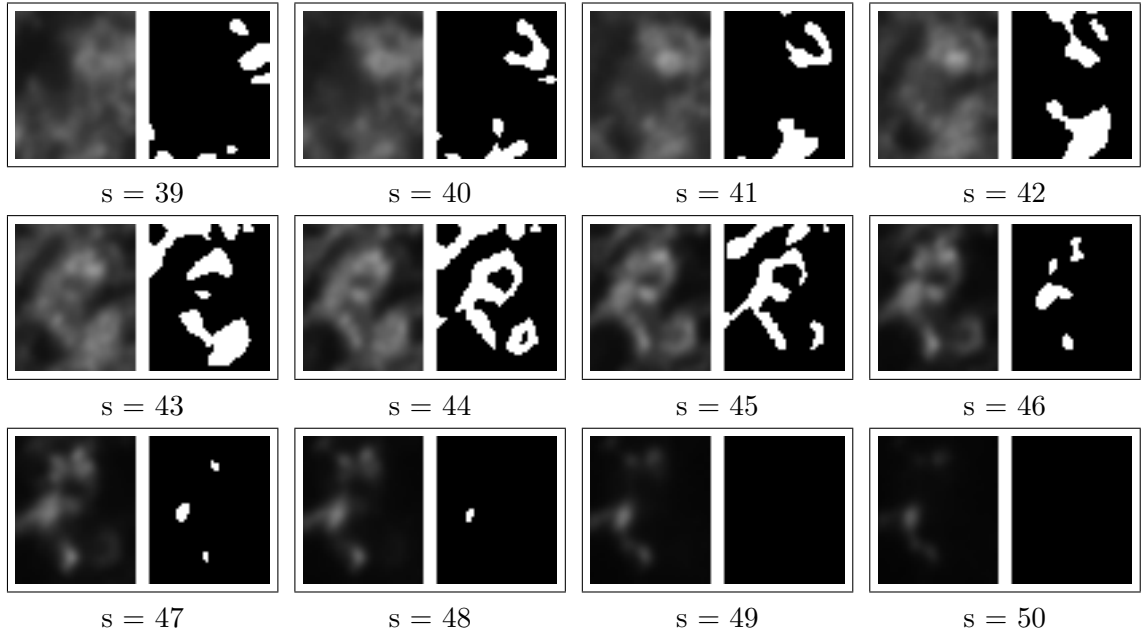


Figure 5.12: Results of Sample 1. *The sample shows the effectiveness of frame prior by detecting the in-focus features with ‘blob’ appearance as connected synaptic features in slice 44 and 45.*

instance, the feature detected by Synapcut at the bottom of the image in Fig. 5.16(b) but is not segmented by the biologists in Fig. 5.16(a) is actually part of the thick membrane of the synapses as shown in the 3-D model in Fig. 5.16(d). Due to the fact that the connection between the feature the rest of the synapses crosses many slices, it is not classified as the synaptic feature by the manual detection when the work is time consuming and the decision is only based on current slice of the considering features. In contrast, the connection with long inter-slice distance is detected by Synapcut due to the frame prior derived from the 3D Laplacian pyramid, which can extract the connectivity of the features through the whole stack.

In order to achieve optimum segmentation results, the likelihoods can be adjusted by adding different weighting parameters, so that the ‘ratio’ between the wavelet-likelihood and the global intensity likelihood can be tuned for the image stacks with severe irregular brightness problem, or the ‘judgement’ of out-of-focus feature can be changed. The smoothness ratio λ in the frame prior can also be adjusted in order to vary the strength of the connection between the synaptic objects. Based on observation, the parameter values outlined in *Section 5.5* were found to give the best result over the entire stack.

However, it is difficult to evaluate the performance of the segmentation. This is because that the manual classification of in-focus features requires the consideration of the different projections of the same image feature at different adjacent slices in the stack which again is very difficult and subjective. Since the parameter adjustment for Synapcut is based on manual evaluation, it is difficult to determine optimal values of the model parameters.

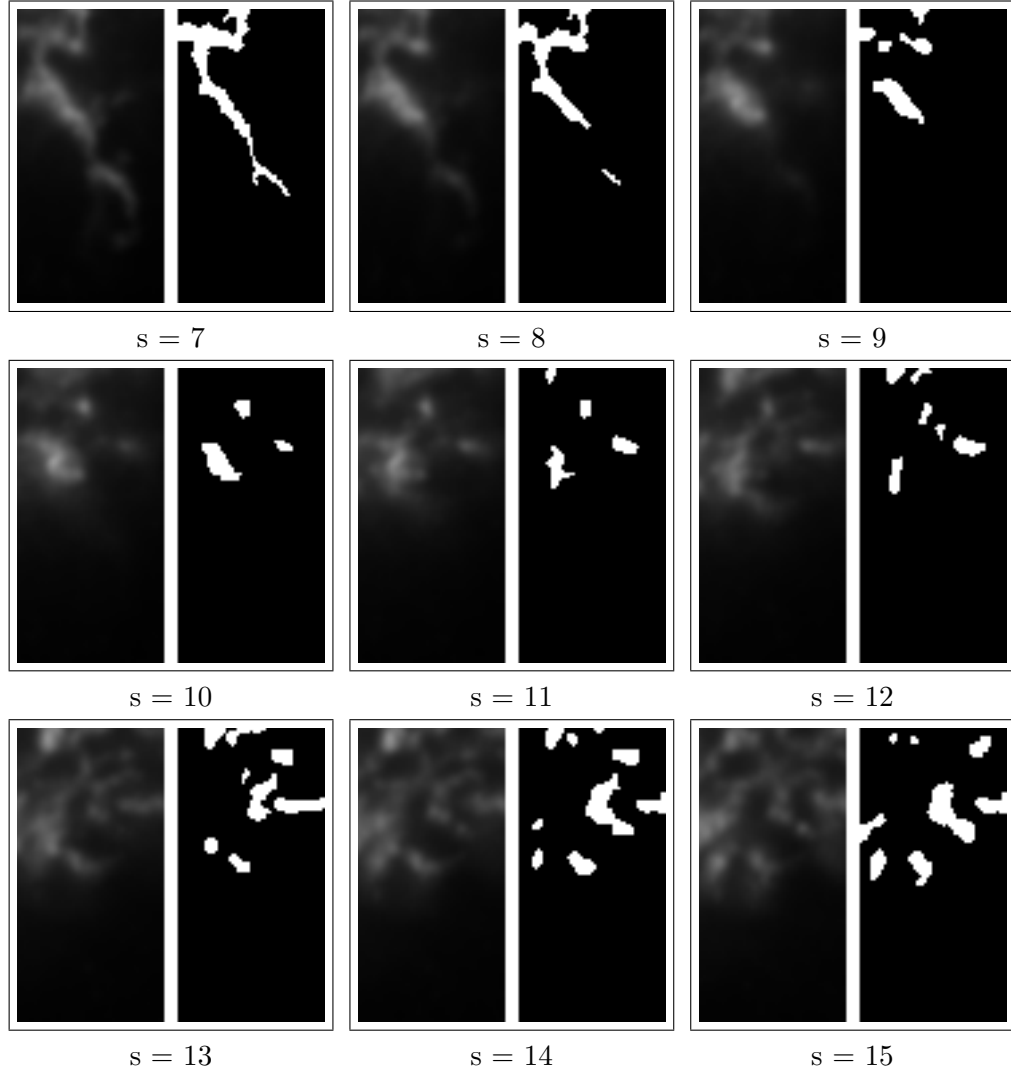


Figure 5.13: Results of Sample 2. *The left part of each image pair is the original image sample extracted from the corresponding slice ‘s’ of the image stack. The right part is the segmentation results of the left images. Through the slices, we can see that the both the bright and dim in-focus features are detected.*

5.7 Conclusion

This chapter presents a Bayesian model for segmenting 3D synaptic objects from a given stack of specimen image slices captured using a fluorescent confocal microscope. The key intra-slice wavelet-likelihood provides a degree of luminance invariance in order to detect the synaptic features with irregular brightness. The other core likelihood based on the vertical inter-slice intensity profile enables the model to distinguish between in-focus and out-of-focus features. Furthermore, a novel frame prior is introduced to direct spatial smoothness along the frame of the synapses and enhance the 3D connectivity of the synaptic features through the whole

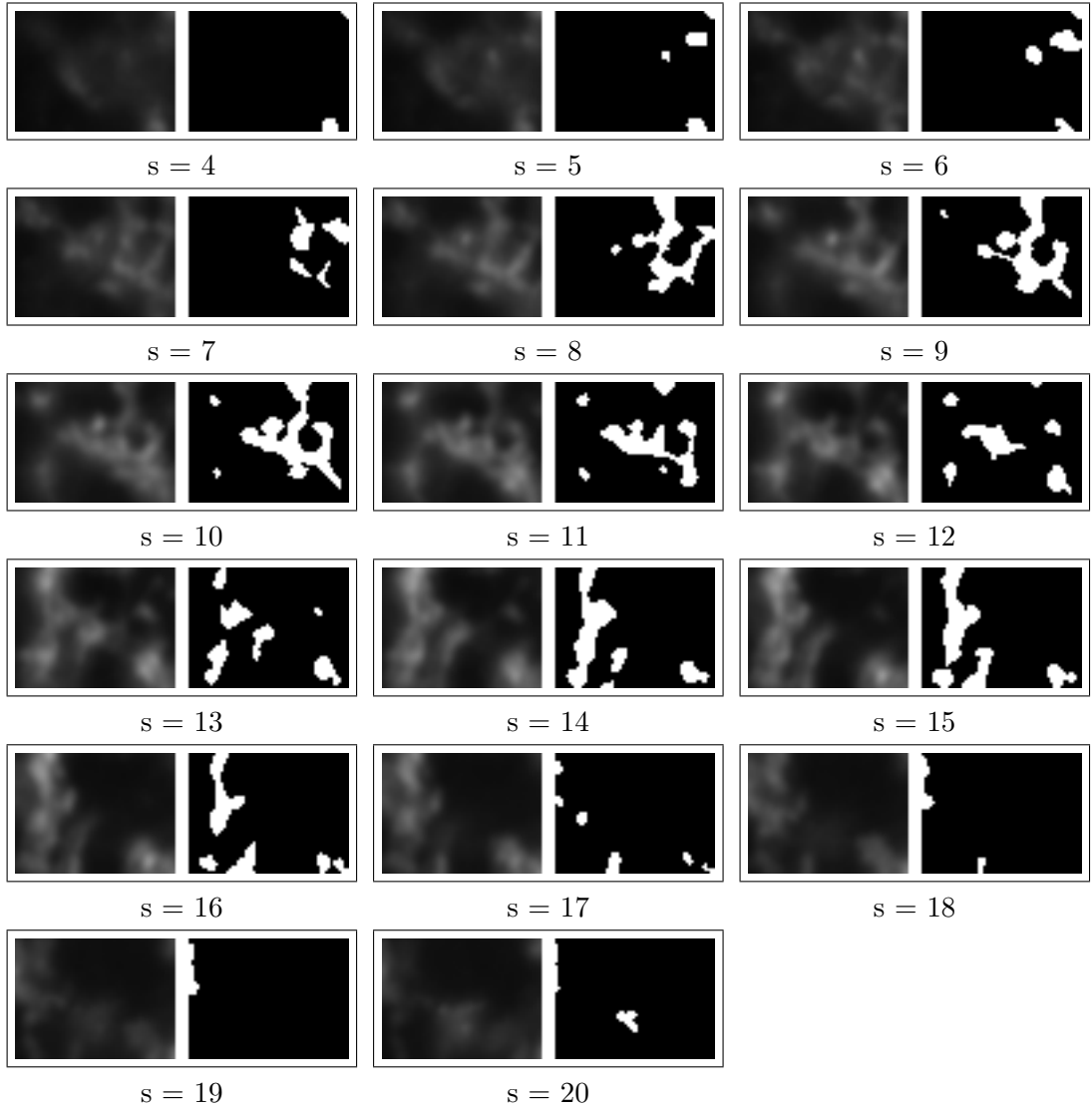


Figure 5.14: Results of Sample 3. *This sample shows that the proposed model can distinguish between in-focus and out-of-focus features. However, the in-focus constraint is too strict for the features in slice 17.*

stack. The complete model (Synapcut) detects the confident in-focus features and keeps the connectivity of the features, and hence provide an approximation of the in-focus synaptic structure. However, the evaluation of SynapCut is limited due to a lack of different image stacks of synapses. Also, due to the difficulty of precise manual evaluation of the segmentation, the choice of the weighting parameters for the likelihoods and prior in the model is a concern in the algorithm. Consequently, further investigation is required for validating the model and finding the optimal model parameters.

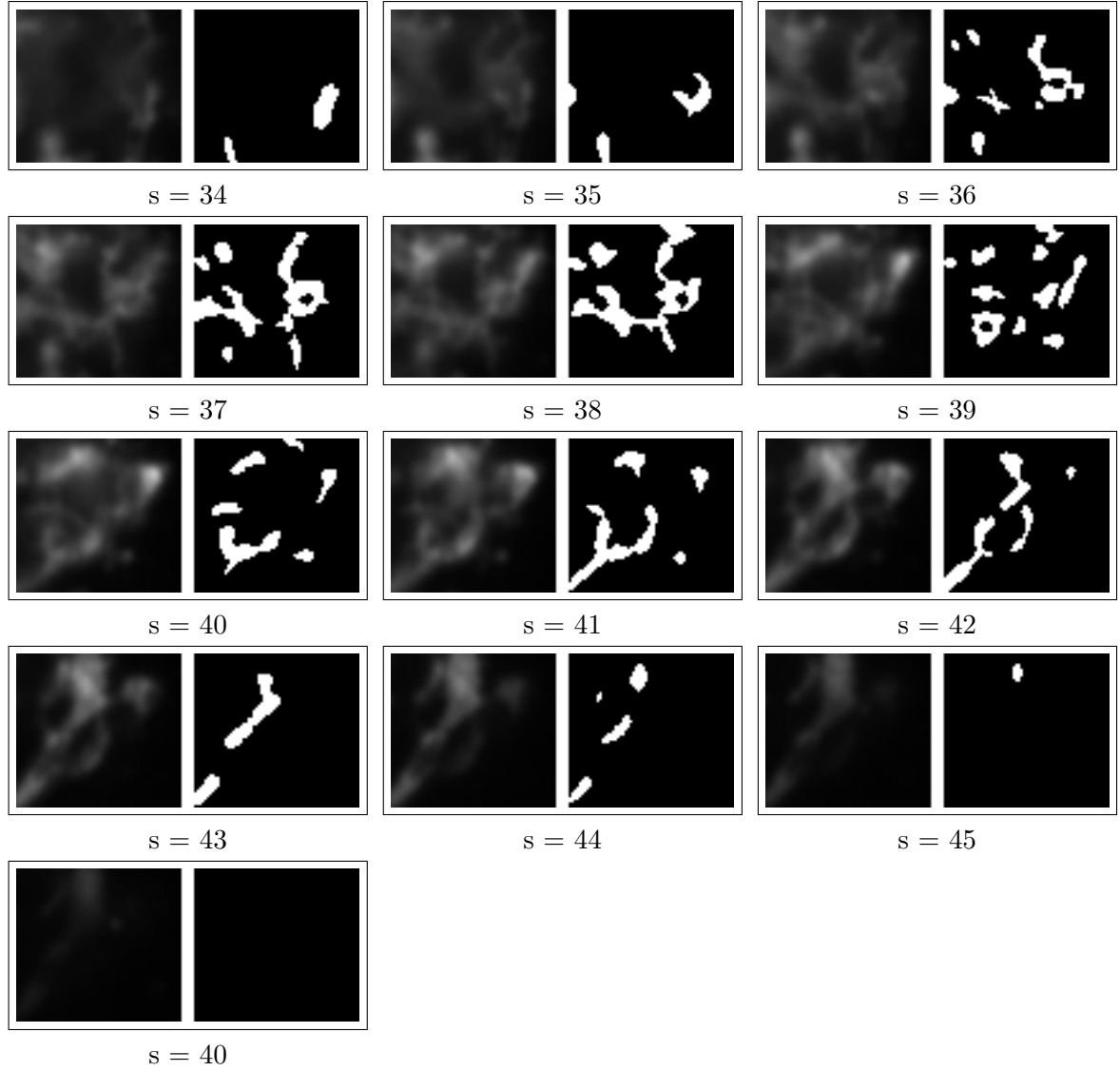


Figure 5.15: Results of Sample 4. *This sample also shows that the proposed model can detect in-focus features while the out-of-focus features are also presented. However, the in-focus constraint is too strict for the features in slice 36 – 39&44.*

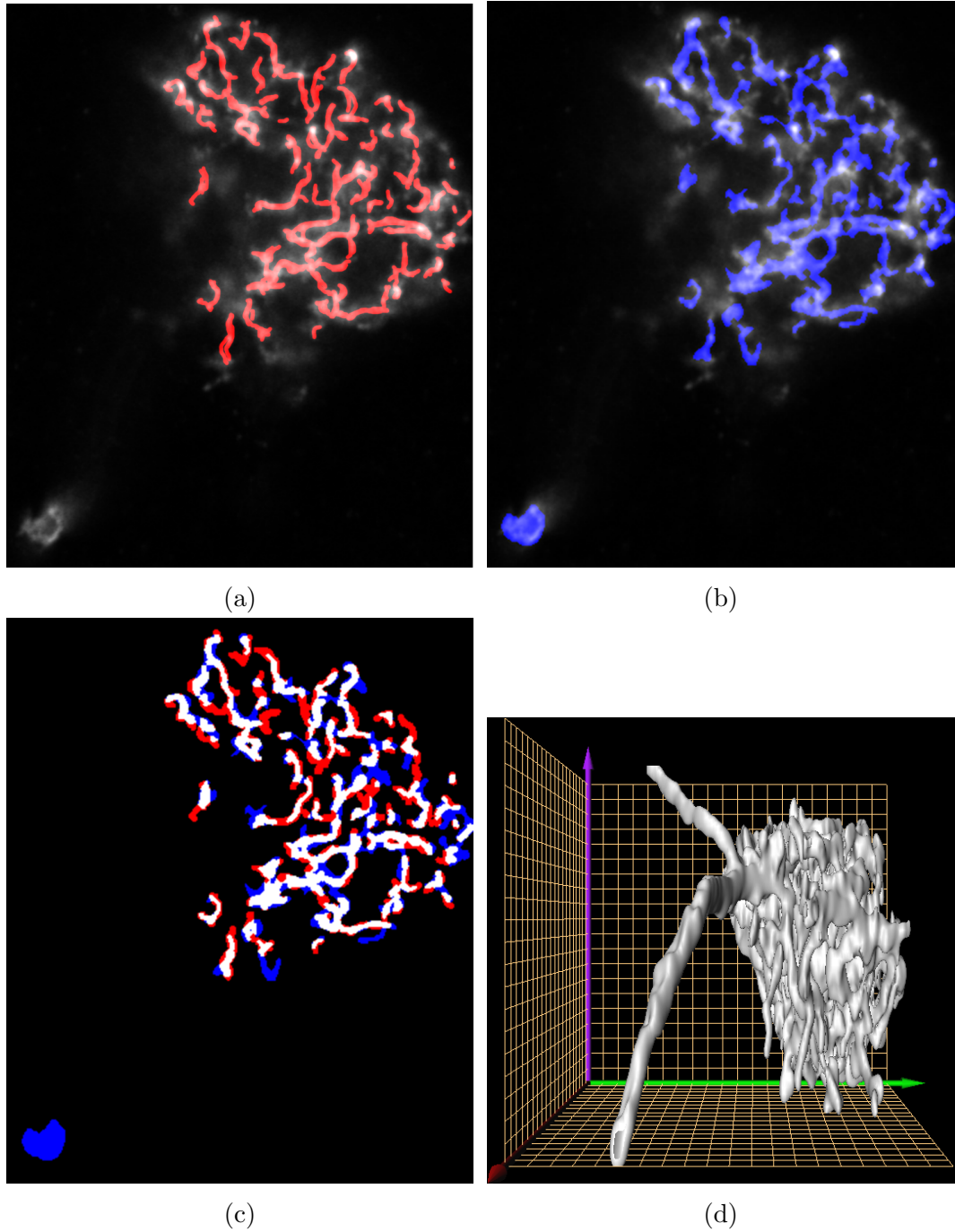


Figure 5.16: Comparison with manual segmentation. (a) a manual segmentation of a slice in the image stack by biologists which is highlighted as red on top of the original image. (b) the segmentation from Synapcut highlighted by blue on top of the original image. (c) the comparison between (a) and (b), where the white indicates the common area, the red highlights the regions in the manual segmentation from (a) that are not detected using Synapcut, and the blue highlights the regions from Synapcut that are not labelled in the ground-truth. (d) a 3D illustration of the segmentation of Fig. 5.2(a) by using Synapcut.

6

Cellsnake: an active contour model for cell/fibre segmentation¹

A novel muscle regeneration technique that uses electrical stimulation to control the growth of the muscles is under investigation for skeletal muscle therapy [40]. In order to validate and evaluate the performance of the new technique, the cell and fibre (formed by multiple cells) objects need to be detected and classified from the image data collected from experimental samples and hence used for further analysis. The segmentation and classification of the cell and fibre objects in the large amount of image data generated would be inefficient by manual analysis from the biologists. Instead, an image processing tool is a more efficient and preferable approach for the image data analysis. The image processing in this work involves segmenting the cell/fibre objects from the given images. However, for the cell body segmentation, when the cells have a regular simple elongated structure, the fibres have varying shapes and have nuclei distributed randomly over the cell body. More importantly, since the observed images can only represent the 3D test samples in a 2D domain, the actual cells/fibres distributed at different depth levels in an observed sample may be projected as overlapped objects in the resulting image, and these overlapped objects with various shapes can ‘confuse’ the classification of individual fibres. In order to distinguish the fibres from overlapped objects and segment the candidate cell/fibre from each overlap object, an active contour model, designed from the ‘apparent’ shape characteristics of the cells, is introduced in this chapter.

¹This work is presented in part in [79]

6.1 Biological Background

The development of new tissue engineering practices for skeletal muscles (i.e. the muscles that are attached to bones) is critical for the reconstruction of lost or damaged muscle. The current treatments for the muscle replacement involve transplanting muscle taken from another part of the body, and the cell based therapies where muscle precursor cells (stem-like cells) are injected into the area to regrow the muscle. However these approaches are yet to be effective. Because muscle cells are responsive to electrical stimulation, a novel technique was proposed in [40] which attempts to use electrical stimulation with conductive polymer materials to control the growth of muscle cells.

The research tries to use muscle tissue engineering and regeneration, mainly in an artificial environment that is outside the injured area, to grow muscle fibres in the lab, or enhance the cell numbers of muscle precursors for cell-based therapies. The research attempts to use electrical stimulation through conductive polymers to control the development of the muscle. The conductive polymers can be coated onto surfaces or existing electrodes and then the cells are seeded onto the polymer and are subjected to an electrical stimulus via the polymer. Electrical stimulation can affect processes at the cell membrane, like the opening and closing of ion channels, and are thought to play a role in controlling cell growth. The muscle fibres that are grown in an artificial environment can then be implanted into the body. This technique can be extended by implanting the conductive polymers in the form of a fibre structure and regenerating the growth process within the injured area. In this case, the polymer fibre would act as a scaffold to support cell growth and itself also provide mechanical strength and function.

For the data analysis, the cells are grown on these polymer substrates and the muscle cell density is measured quantitatively in order to assess the performance of the various processes of this technique. Fig. 6.1 shows a typical image (1040×1388) that superimposes the images of both cells and fibres from these experiments. The red objects are the cell bodies of the cells/fibres and corresponds to the cell body image in Fig. 6.2(a) and the blue ‘blobs’ that are inside red objects are the nuclei that corresponds to Fig. 6.2(b). A red object is classified as a cell or fibre based on the location and amount of nuclei (blue “blobs”) inside the object. When more than one nucleus exists in the same ‘object’ that object is a fibre, otherwise it is a single cell. In order to indicate the degree of cell growth (increase in cell numbers) and how many cells go on to form fibres (fibre number and multi-nucleate density), the analysis of this research focuses on counting the number of nuclei and fibres as well as the ‘fibre/cell density’. The ‘fibre/cell density’ is the ratio between the amount of fibres and the cells in the specimen which are estimated by counting the number of the two different types of objects from the images.

The image processing goal is to detect and classify the objects in a given image as single cells or fibres, and hence count the number of nuclei in the ‘classified’ fibres. Since the cell bodies and the nuclei are imaged at the same time from a fixed testing sample, the observed images of the nuclei and cell bodies are registered. Consequently, the segmentation and classification of

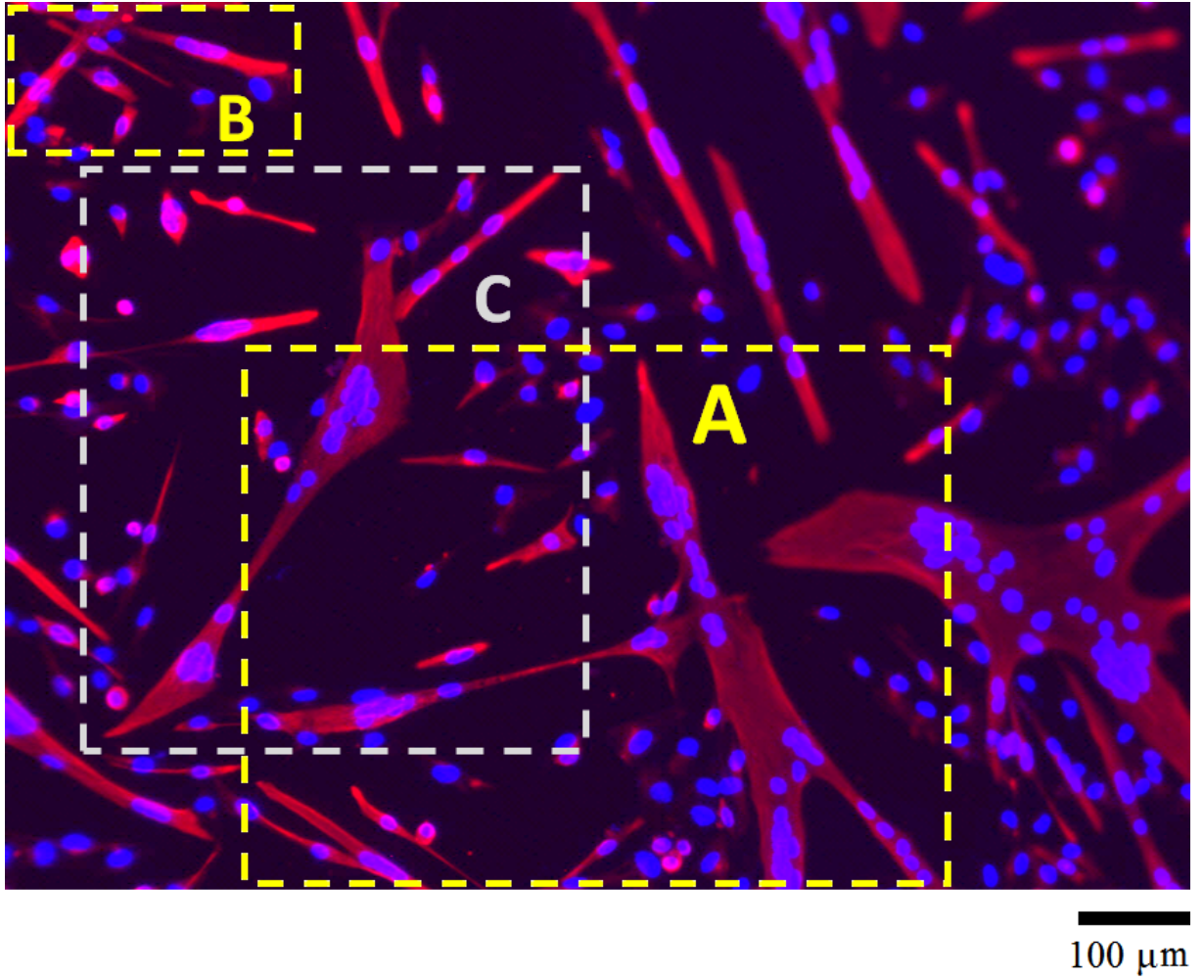


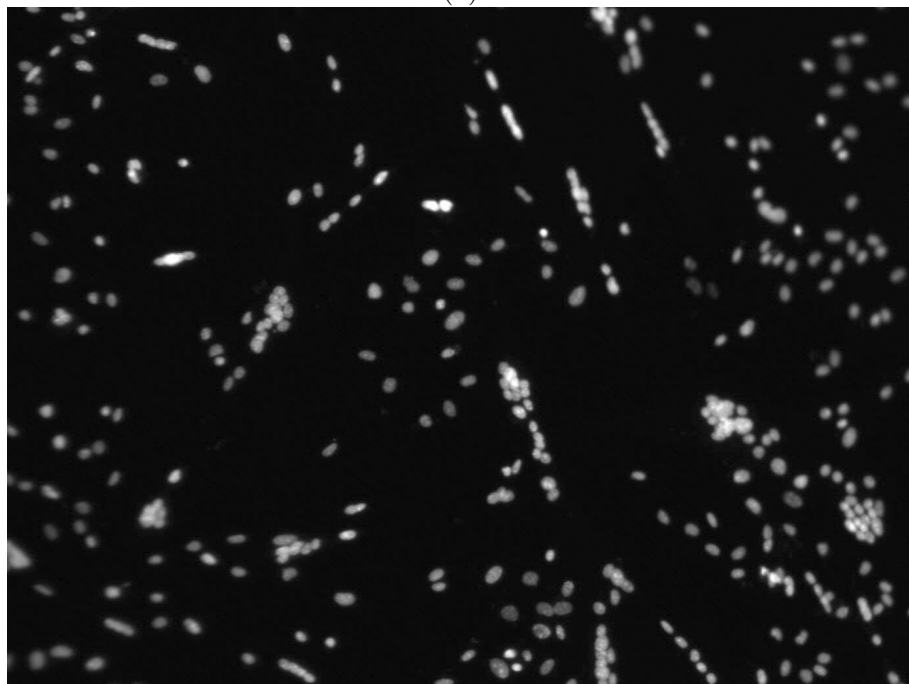
Figure 6.1: Sample of skeletal cells/fibres. *(The red objects are the cell body of the skeletal cells and the blue spots are the nuclei of the cells. The highlighted rectangular regions are the reference regions for the results presented in the later sections.)*

the cell and fibre objects involve segmenting the cell body objects from the cell body image and counting the number of nuclei in the nuclei image for the corresponding segmented cell bodies. A manual segmentation and classification for Fig. 6.1, i.e the ground-truth, is shown in Fig. 6.3, where the cells are highlighted by ‘yellow’ contours and the fibres are highlighted by ‘green’ contours. Also, the individual nuclei are highlighted by the small ‘blue’ circles with ‘white’ fill.

For counting the nuclei, as shown in Fig. 6.2(b), some nuclei are clustered and given merged appearance in the image which makes the enumeration of the nuclei difficult. This enumeration problem is similar to the spot detection problem discussed in *Chapter 3*. Intuitively, the spot objects in the nuclei image can be fitted with Gaussian mixture models through the eSMEM shape modelling algorithm. This allows the nuclear clusters to be parameterized and hence the



(a)



(b)

Figure 6.2: Cell bodies and nuclei of the skeletal cells/fibres. ((a) and (b) show the cell bodies and the nuclei respectively of the skeletal muscle sample in Fig. 6.1, which are corresponding to the red objects and the blue ‘blobs’ respectively.)

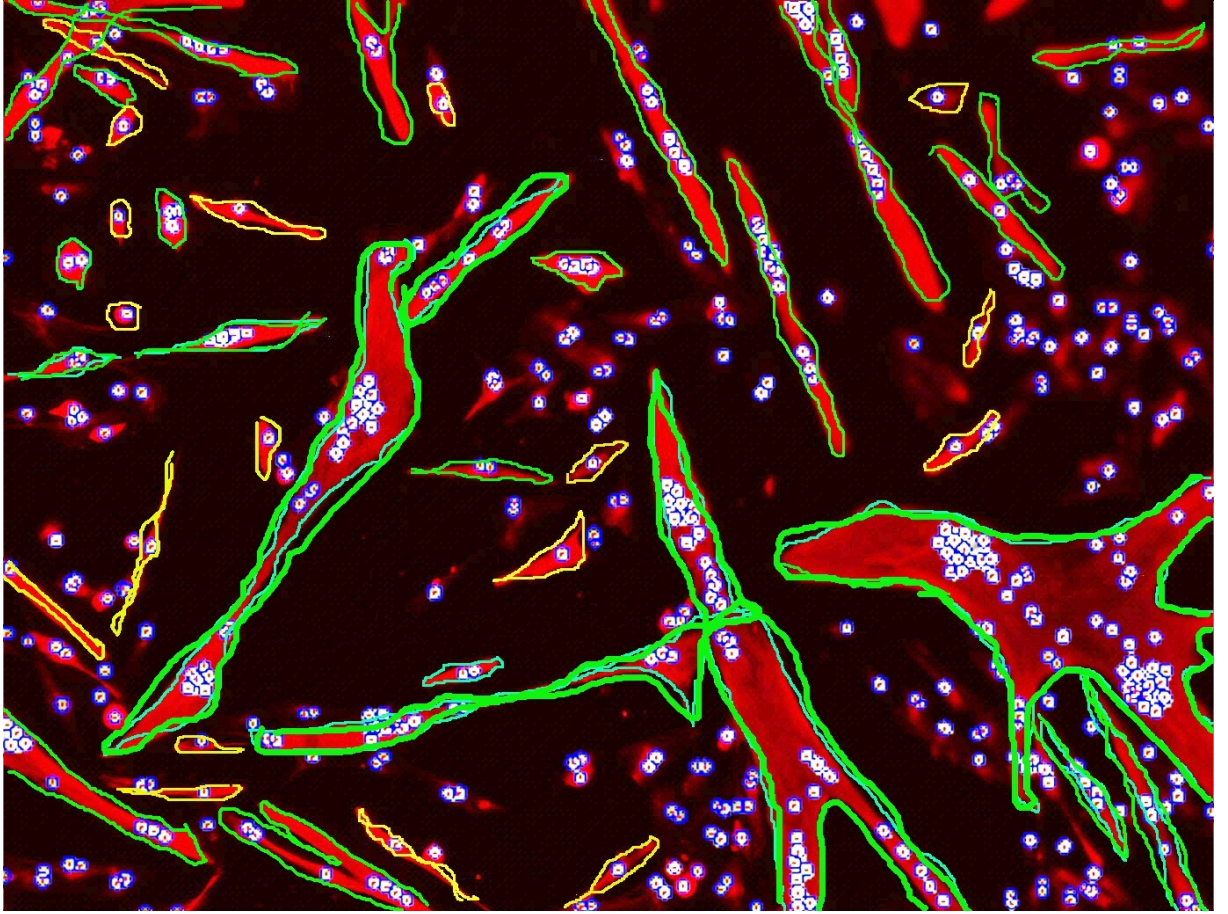


Figure 6.3: Manual segmentation. *(The white objects are the nuclei, and are separated into individual nucleus by the blue circles. The objects highlighted by yellow line and green line are the single cells and fibres respectively. Obviously, a single cell has only one nucleus and a fibre has multiply nuclei, which can be a cluster of nuclei or even multiple clusters of nuclei.)*

numbers of candidate nuclei are appropriately estimated. For the cell body segmentation, the objects as shown in Fig. 6.2(a), are usually well separated in the images, but some, as shown in Fig. 6.4, appear to ‘touch’ or are overlapped. The overlap appears because the image is only a 2D projection of the testing sample with essentially an arrangement of cells in 3D. Therefore, the goal of the cell/fibre object detection is to segment the ‘joined’ or overlapped objects before classifying such objects.

As shown in Fig. 6.2(a), the overlapped objects are rarely completely overlapped in the image. This means the ‘branches’ or non-overlapped portions of each (red) region (see A, B and C in Fig. 6.1), can be used to infer the estimation of the occluded parts of the objects *through* the occluded regions. Intuitively, this idea is accomplished by the active contour technique to segment the overlapped objects, where the non-overlapped portions of the objects (cells or fibres) infer the development of the contours through the overlap region. However, the existing

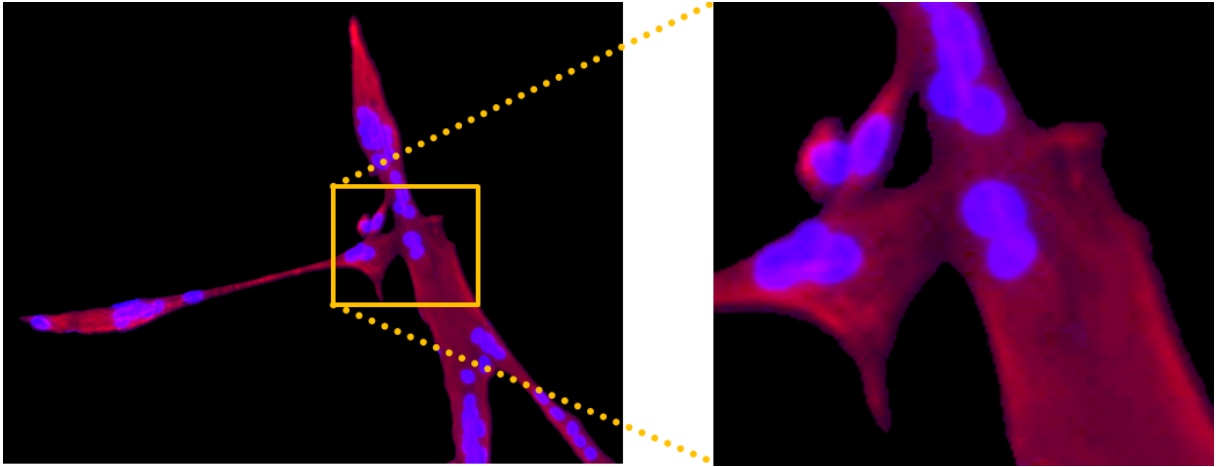


Figure 6.4: Overlap problem. *(This is a sample of the overlapped objects in the region ‘A’ of Fig. 6.1. Since there is no transparency available in the overlap region, the occluded object has no information available in such region. Hence the topology information is ambiguity for the occluded object while doing the segmentation.)*

active contour models are not designed for segmenting the overlapped objects, especially when the prior shape information of the fibre objects are difficult to define, as the shapes of the fibres are highly irregular. In order to segment the overlapped objects, shape information plays an important role in regulating the deformation of the active contours in the occluded portions. In order to define the shape regulation and to link these ideas together, a new active contour model named Cellsnake is introduced in this work.

Consequently, the idea of solving the task of counting the cell and fibre objects in a skeletal muscle image is to combine Spotnick and Cellsnake as a joint decision, where Spotnick is used to count nuclei with GMMs and Cellsnake delineate the object boundaries for both overlapped and non-overlapped cell body objects using novel forces and a novel merge scheme to overcome the overlap problem. By joining the estimations of the two parts, each estimated cell body object is classified as cell or fibre based on the number of estimated nuclei inside the object, and the biologists can then access the fibre/cell density based on the numbers of the estimated cells and fibres from the image.

Before explaining the detail of the Cellsnake, a review of the active contour models is first given in *Section 6.2*. Based on the knowledge of the active contour models, the Cellsnake algorithm that has adopted the ideas of several existing techniques is described and discussed in *Section 6.4*. The results of the Cellsnake on segmenting the artificial ground-truth data and the real image data are presented in *Section 6.5*.

6.2 Active Contour

The active contour (AC) model, also called a ‘snake’, is a popular deformable contour model for shape analysis in image processing. The use of the snake to extract features (or objects) of interest in images was originally introduced by Kass *et al.* in [54]. The idea of the AC model is to deform a continuity spline, i.e. a contour, iteratively with designed regulations until the process meets a particular termination criterion so that the contour fits to the ‘boundary’ of the desired object in image. Also, the AC can be described as a contour development model that is regulated by different ‘forces’ derived from different aspects of the image data and different external constraints. In an image, a snake is represented by a parametric spline formed by the control points S with the corresponding positions $\vec{u}_S = \{\vec{u}_1, \dots, \vec{u}_s\}$ at time or iteration t during the deformation, where $\vec{u}_s = (h_s, k_s)$ is the position for the control point s . The control points S are referred to as “snaxels” in image processing since S can be considered as the ‘elemental particle’ of the snake. The deformation of the AC is governed by a continuous energy minimization where the energy function of the model is the sum of the energy functional defined by the different forces acting on each snaxel S and is expressed, according to [54], as

$$\begin{aligned} E_{snake}^* &= \sum_S E_{snake}(s) \\ &= \sum_S E_{smooth}(s) + E_{con}(s) \end{aligned} \quad (6.1)$$

where E_{smooth} represents the inherent energy of the contour due to bending and can be considered as the internal forces that maintain the smoothness of the AC. E_{con} is the conditional energy that gives rise to the image forces and the constraint forces for regularizing the deformation of snake under various conditions. The images forces push the snake toward salient image features like lines, edges, and subjective contours, while the constraint forces regulate the deformation of the snake based on specific properties of the image objects known a priori. The final position of the AC corresponds to the equilibrium of the forces which is reached at the minimum energy of the model.

The internal smoothness energy of the snake maintains the elasticity and rigidity of the contour during the deformation and defined as

$$E_{smooth}(s) = \frac{\alpha(s)|\vec{u}'_S(s)|^2 + \beta(s)|\vec{u}''_S(s)|^2}{2}$$

where $\vec{u}'_S$ is the 1st derivative of $\vec{u}_S$ with respect to s and the $\vec{u}''_S$ is the 2nd derivative. The 1st-order term and the 2nd-order term of the spline are controlled by the weights α and β respectively. Note that the 1st-order term of E_{smooth} makes the snake like a membrane in which the snaxels would tend be stretched if the adjacent snaxels are stretched. The sensitivity of the snaxel s to the stretch is controlled by $\alpha(s)$. The 2nd-order term makes the snake act like a thin plate so that the ‘sharp’ bending of the spline would be smoothed or the curvature in

the spline would be reduced. This ‘smoothening’ effect is controlled by $\beta(s)$ at s . In the other words, α and β determine the elasticity and rigidity respectively of the AC model.

6.2.1 Numerical Solution

To find the shape of the AC, i.e. the positions $\vec{u}_s$ of all of the snaxels S , where the energy E_{snake}^* of the model reaches the minimum, the equation in Eq. (6.1) is differentiated with respect to $\vec{u}_s$ and becomes

$$-(\alpha \vec{u}_s')' + (\beta \vec{u}_s'')'' + \nabla E_{con}(S) = 0 \quad (6.2)$$

In order to solve this equation analytically, the derivatives are approximated by a finite difference. Also, by rewriting the derivative $\nabla E_{con}(S)$ of the conditional energy as the sum of conditional forces $F_{con}(S)$ so that $\nabla E_{con}(S) = \vec{F}_{con}(S)$, the equation in Eq. (6.2) becomes

$$\begin{aligned} & \alpha(s)(\vec{u}_s - \vec{u}_{s-1}) - \alpha(s+1)(\vec{u}_{s+1} - \vec{u}_s) \\ & + \beta(s-1)(\vec{u}_{s-2} - 2\vec{u}_{s-1} + \vec{u}_s) \\ & - 2\beta(s)(\vec{u}_{s-1} - 2\vec{u}_s + \vec{u}_{s+1}) \\ & + \beta(s+1)(\vec{u}_s - 2\vec{u}_{s+1} + \vec{u}_{s+2}) \\ & - \vec{F}_{con}(s) = 0 \end{aligned} \quad (6.3)$$

This can be written in the matrix form

$$\mathbf{A}\mathbf{u}_S + \mathbf{F}_{con} = 0 \quad (6.4)$$

where $\mathbf{A}$ is a pentadiagonal matrix, $\mathbf{u}_S$ is a vector of the positions $\vec{u}_s$ of the snaxels and $\mathbf{F}_{con}$ denotes the vector of $\vec{F}_{con}(\cdot)$ at the snaxels. The goal of the AC model is to find a solution for Eq. (6.4) which gives rise to a good contour for the desired image object. In practice, the deformation of the snake is equivalent to iteratively updating the positions $\vec{u}_s^t$ of the snaxels during the deformation process, where $\vec{u}_s^t$ denotes the positions of the snaxels at iteration t . The evolution of the positions stops when the AC model reaches the minimum energy. This means the derivative of $\vec{u}_s^t$ with respect to the time t , i.e. iterations, would become 0 when the derivative of the energy function with respect to $\vec{u}_s^t$ becomes 0. Hence, Eq. (6.4) becomes

$$\mathbf{A}\mathbf{u}_S^t + \mathbf{F}_{con} = 0 = \frac{\partial \vec{u}_S^t}{\partial t} \quad (6.5)$$

Again, by approximating $\frac{\partial \vec{u}_S^t}{\partial t}$ with finite difference, Eq. (6.5) becomes

$$\begin{aligned} \mathbf{A}\mathbf{u}_S^t + \mathbf{F}_{con}(\mathbf{u}_S^{t-1}) &= -\gamma(\mathbf{u}_S^t - \mathbf{u}_S^{t-1}) \\ \mathbf{u}_S^t &= (\mathbf{A} + \gamma\mathbf{I})^{-1}(\mathbf{u}_S^{t-1} - \mathbf{F}_{con}(\mathbf{u}_S^{t-1})) \end{aligned} \quad (6.6)$$

where γ is the step size of snaxels during the deformation, and $\mathbf{u}_S^t$ is a vector of the positions $\vec{u}_s^t$ at the t^{th} iteration in the evolution. The matrix $(\mathbf{A} + \gamma\mathbf{I})$ is a pentadiagonal banded matrix

and the inverse matrix $(\mathbf{A} + \gamma\mathbf{I})^{-1}$ can be calculated by the LU decomposition. This matrix accounts for the smoothing effect of the internal energy on the snake. Eq. (6.6) is the evolution equation for the AC model that updates the positions of the snaxels at each iteration of the deformation process. The snaxels become stationary when the model reaches the equilibrium of the forces. Hence, the iterative process is practically terminated when the positions of the snaxels are unchanged or have relatively small change during the deformation.

6.2.2 A Review of the Existing Conditional Forces $\mathbf{F}_{con}$

As shown in Eq. (6.6), the conditional forces $\mathbf{F}_{con}$ are the key to attract the snake to the desired image features under various conditions. In [54], Kass *et al.* describe several image forces derived from the line, edge and termination features in the image which are referred to as the external energy for the AC model. The line functional attracts the snake to light lines or dark lines in the image based on the intensity of the features. The edge functional attracts the snake to the edge features with large image gradient. The termination functional attract the snake to the broken edges.

The performance of the AC models for fitting the shapes of different image objects depends on how well the conditional forces are related to the features of the objects. Since the snakes generally provide locally minimized solutions, the different conditional forces proposed by the existing alternative AC models have the purpose of achieving initialization invariance and robust convergence for the snakes.

The force derived from the image gradient can attract the snake to the edge features in the image. However, images with noise or image objects that have rough intensity appearance, i.e. contain weak edges inside the objects, can cause failure of the fitting. To solve this problem, Cohen introduced a pressure force in [25] that pushes the snaxels in the directions of the normals to the corresponding curves on the AC and hence ‘inflates’ the AC. In this case, the snake behaves like a balloon which is blown up by the pressure force, and with the help of the internal smoothness forces, the AC can pass through the noise and weak edges and is stopped by the strong boundary, i.e. step edges of the desired image object. This approach modifies the image forces (derived from gradient of the image) presented in traditional Kass *et al.* snake. Moreover, with the pressure force, the initial snake need not be too close to the object boundary to converge.

The parametric gradient vector flow (GVF) that is introduced by Xu and Prince in [93,94] has been widely used in active contour models to give rise to conditional forces. The GVF is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image, or the gradient vectors of the color image in L, U, V channels proposed by Yang *et al.* in [96]. In this case, the edge gradient field is extended and smoothened so that the snakes are attracted to the broken object edges and concave object boundaries. This also means the GVF can attract the snakes to the edges even if the snakes are initialized away from the edges. Hence, the GVF snake is more robust to initialization than the Kass *et al.* model.

Alternatively, Li and Acton proposed a constraint force using vector field convolution (VFC) in [58] that extend the vector flow to large region. The VFC convolves the image edge map with a vector field kernel, which comprises radial symmetric vectors pointing towards the center of the kernel. This is equivalent to assigning an attraction force for each pixel position based on which the snake evolves. With the VFC, Li and Acton have shown better initialization invariance and noise insensitivity for the AC model in [58].

Different to the GVF snake and VFC snake, Caselles *et al.* introduced the geodesic snake model in [22] to improve the initialization invariance of the Kass *et al.* model for the objects with large homogeneous regions. The geodesic snake model is based on the flow that is derived from the inverse of the gradient of a ‘smoothed’ version of the image. Based on this flow, each point on the contour moves in the direction of its normal at a speed proportional to the magnitude of the flow vector. Hence, the image dependent constant flow shrinks or expands the contour towards the object boundary.

In [80], Paragios et al. integrated the GVF with the geodesic AC model, referred to here as GeoGVF. The GVF provides the bidirectional force to drive the contours towards edges, while the constant flow from the geodesic model poses conditional inflation or deflation forces on the contour. The additional adaptive bidirectional constant force is designed to determine the curve propagation when the GVF term becomes inactive, e.g. at saddle points.

Shape concavity is a difficult issue for the propagation of the snake. This is due to the confusion of the GVF in narrow concave regions. Also, the internal smoothness functional resists the snake from forming concave curves. To ease this problem, Gil and Radeva proposed a distance transform in [39] using a modified mean curvature flow. The curvature vector flow is computed from this distance transform to attract contours into shape concavities. The idea was to evolve the object boundaries according to curvature flow till they were no longer concave, where the inward or outward propagation was defined by non-negative or non-positive curvature flow.

Furthermore, the level set method proposed by Osher and Sethian in [74] is a widely used technique to implicitly represent active contours that handle topological changes [22,23]. When the image objects have ‘rough’ appearance, which are also called topological issues, it is difficult to determine the evolution of the contour and the re-parametrization of the contour after every topological change. To solve this problem, the snake is embedded in the zero level set and the temporal evolution is achieved by deforming the entire level set function. Hence, the snake can naturally model contour propagation and topological changes. In the level set formulation, Chan and Vese developed a popular region-based AC model in [23] based on the Mumford-Shah functional for segmentation [69]. The Chan-Vese active contours do not depend on the image gradient but are instead related to a particular segmentation of the image by minimizing both the area functionals inside and outside the AC, and hence is robust for noisy image objects.

Also, some active models integrate the shape priors of the target objects to the conditional forces in order to deform the model to fit complicated shapes based on a priori knowledge of the

object shapes. For instance, the ASMs and the AAMs are popular models for extracting the boundary of complex objects by using statistical models trained from multiple reference images of the same object or similar objects, where the ASMs usually use the prior shape information of the object and the AAMs use both shape information and the inner texture information of the objects. A similar idea is also used by the multi-reference AC model proposed by Foulonneau *et al.* in [38], where the shape information of each reference object is encoded by a set of geometric moments (i.e. the shape descriptors) of the inner region of the shape. This model can apply multiple sets of geometric moments for segmenting different objects at the same time. Also, based on the shape constraints of the geometric moments, the model can segment the overlapped objects of different types in the image and is robust to the initialization.

6.2.3 Discussion

The task in this part is to determine whether a separate region with complex shape in this image is a single fibre or consists of multiple overlapped fibres, and segments the overlapped fibres. The difficulties of this task are the highly varied shape appearance of the fibres and the shape estimations of the occluded parts of the fibres. The multi-reference AC proposed by Foulonneau *et al.* in [38] may be a solution for segmenting the overlapped candidates. However, finding the reference fibre shapes for the model is not feasible when there is no regularity or similarity in the fibre shapes and at the same time the overlapped candidates have similar surface topology (i.e. similar inner texture) in the image. This also applies to the active models that are based on reference objects, such as the ASMs and the AAMs. As the shapes of the objects are so varied, it is challenging to come up with a representative set of shapes from training data. This makes training an ASM or AAM difficult. Since the final goal is to count the number of cells and fibres in the image, which are differentiated by the number of nuclei inside the cell bodies, the aim of the segmentation is to find the approximate shape of each candidate of an overlapped object that ‘covers’ the correct nuclei (which are parameterized by GMM fitting using Spotnick) for the corresponding cell/fibre, rather than getting the exact boundaries of the cells/fibres.

Most of the existing alternative AC models as described in the previous section and in *Section 2.1.5 in Chapter 2*, focus on finding the exact boundaries of the target objects in different images conditions. Also, compared to the fibres in this task, the objects encountered by those models either have less shape variability or the shapes are less complex. In this case, the decision of whether a ROI is a single fibre or a group of overlapped fibres/cells is made by considering how the non-overlapped parts in the ROI are ‘connected’ through the overlap regions. As shown in Fig. 6.5, the circled fibres obviously span the overlap region as their axes are almost colinear. However, the three circled regions are considered as separate fibres because their orientations are almost perpendicular.

There are two ways to model this idea with ACs. The first approach is a conventional approach that assigns one AC to each ROI (non-overlap or overlap) and split the AC to fit the

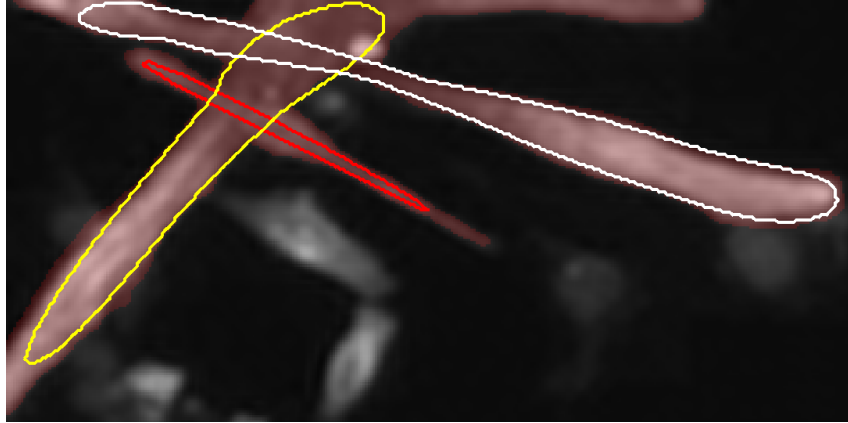


Figure 6.5: ROI with overlapped fibres/cells. (*The ROI is highlighted in red, and the overlapped fibre/cell candidates are highlighted by lines with different color.*)

overlapped candidates the deformation meets certain split criteria (i.e. the ROI is classified as an overlap region). However, as shown in Fig. 6.2(a), the prior shape information of the objects is difficult to formulate and hence this approach requires some complex constraints in the model to alter the deformation of the AC, and a complicated split scheme is also needed. Alternatively, instead of fitting a single AC to each irregular ROI with a set of complex constraints, the second approach is to use multiple small ACs to fit different parts of the ROI and merge the ACs that belong to the same cell/fibre object. In this case, the ACs are assigned to the non-overlapped parts of the ROI, and merge criteria are based on the orientations of the ACs and the overlap area compared to the total internal area of the ACs. The combination of the merged ACs is then the overall shape approximation for the corresponding fibre candidate. The problem in the second approach is designing the conditional forces to constrain the deformation of the ACs in the overlap area based on the rest of the object shape. In fact, no matter how the shape of a fibre varies, it is yet formed from a group of cells. As shown in Fig. 6.1 & Fig. 6.2(a), the cell objects have simpler and more regular shape characteristics which are thin and elongated. Intuitively, with the merge approach, the small snakes are considered as ‘cell candidates’ in the ROI, and the deformation is regulated by a cellular shape functional. Finally, shape estimates of the overlapped fibre objects are obtained by merging the ACs associated with the real cell/fibre candidates based on the merge criteria. The merge approach is more feasible and preferable than the split approach since the ‘cellular’ shape functional is easier to define than the fibre shape functional when the cells have simpler shape characteristics than the fibres in the image.

Since the goal is to find the approximate shape of each candidate that covers the nuclei for the corresponding cell/fibre, and the clusters of nuclei are located separately inside each ROI, each of the small ACs can be assigned to each separate nucleus cluster in a given ROI, and then merge the ACs when the corresponding nuclear regions belong to the same fibre after the deformation. Accordingly, the snakes are initialized as the boundaries of the isolated nuclear

regions and the evolution is constrained by the ‘cellular’ shape functional. What is needed therefore is to formulate the cellular shape functional. The key of the AC model in this work is the implementation of simple constraint forces derived from the patterns of cellular appearance in the observed image to enforce each snake to deform in the shape of a cell. Since the snake is considered as a ‘hypothetical’ cell, the AC model is called Cellsnake. Although the existing AC models are not designed for the objects in this task, some of the ideas are integrated to the design of the new conditional forces. Also, in order to achieve a good fitting result, Cellsnake incorporates the shape information with the vector field convolution (VFC) technique from [58] and the Geodesic model from [22] to alter the deformation based on the intensity profile of the image. Before exposing the essence of the new AC model, some pre-processing steps for image enhancement, combined with object segmentation and nuclear region segmentation are firstly presented in the next section.

6.3 Pre-processing

The input image, as shown in Fig. 6.2, is formed from a nuclear image and a cell-body image (the cells and fibres) in gray scale, which are superimposed in different channels of the image. Since the fine details inside the cell/fibre regions are not relevant in this work, the object can be smoothened in order to avoid false deformation of the model due to the noise and rough topology. Here, the input image is smoothened by using the wavelet de-noising technique [5] combined with the dual-tree complex wavelet transform [86]. The detail of the wavelet de-noising approach can be found in [5,86], and a sample of the smoothing result is given in Fig. 6.6(c).

Since the non-overlapped cell/fibre objects and nuclear regions are significantly distinguishable from each other in the images, and the exact boundaries of the regions are not of interest, the smoothened images are then simply thresholded to separate the individual regions. The threshold value is set to 50 based on the consideration in this work. These filtered regions are therefore the ROI, i.e. the combined object, for fitting the active contours. Because a nucleus is located approximately at the centre of a cell, it is intuitive to use the boundary of each isolated nuclear region as an initial contour of the corresponding raw fibre/cell region.

However, as shown in Fig. 6.6(a)&(c), the ‘nuclear regions’ in the cell body image are notably dimmer than the surrounding area inside the cell/fibre. This is because the nuclei obstructs those regions and reduces the light emitted from the cell bodies in the occluded regions during the imaging process, and gives rise to less intensity in the resulting image. This imaging defect can cause unwanted deformation of the snake, where the snake will treat these dimmer regions as the boundary area of the ROI. To solve this problem, the original cell body image $\mathbf{I}_{\text{muscle}}$ is simply added with the nuclear image $\mathbf{I}_{\text{nuclei}}$ by a weight λ_{nuclei} (usually is set to 0.4) as follows

$$\bar{\mathbf{I}}_{\text{muscle}} = \mathbf{I}_{\text{muscle}} + \lambda_{\text{nuclei}} \mathbf{I}_{\text{nuclei}} \quad (6.7)$$

$\bar{\mathbf{I}}_{\text{muscle}}$ is the corrected muscle image (Fig. 6.6(d)). It is referred to as $\mathbf{I}$ in the following section

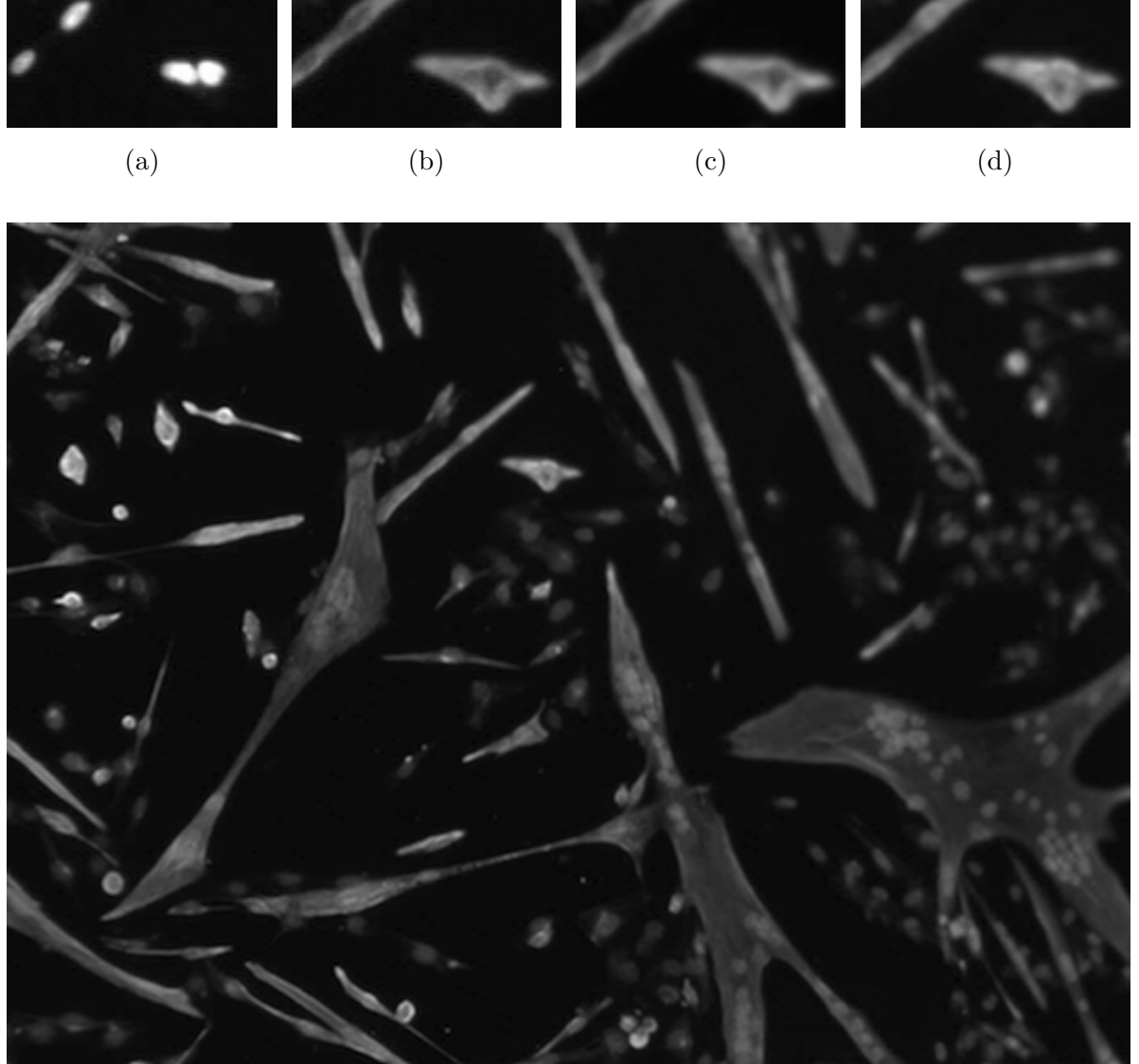


Figure 6.6: Sample of modified composite image of the region ‘A’ in Fig. 6.1. ((a) & (b) are the respective nuclei image and cell body image of region A in Fig. 6.1. (c) is a smoothed version of (b) after wavelet denoising. (d) is a corrected image of (c) by boosting the dimmer nuclei area. The bottom image shows the full size corrected image.)

and provides the geometric information to the deformation of our active model.

The initialization of the ACs is set by the boundaries of the separate nuclear regions in the corresponding ROI. Since the goal of the initialization is to find the number, position and approximate shapes of the nuclear regions in the ROI in order to start the modelling, but not to find the exact boundaries of the nuclear regions, the original nuclear image is firstly smoothed by a small Gaussian filter, and the nucleus clusters are then isolated from the blurred image by thresholding the image with a user defined value (it is set to 40 in this work). As mentioned

earlier, the number of the nucleus candidates in each cluster is estimated by fitting a GMM to the cluster using Spotnick as described in *Chapter 3*. Also, each of the isolated nuclear region is assigned with one snake that is initialized as the boundary of the nuclear region.

6.4 Cellsnake

Recall that a snake is represented by a closed parametric curve formed by control points (snaxels) S with positions $\vec{u}_S^t = \{\vec{u}_1^t, \dots, \vec{u}_s^t\}$ where $\vec{u}_s^t$ is the position of the snaxel s at iteration t during the deformation. The initial snake is refined interactively and attracted towards the object boundary by the constraint forces derived from the cellular shape information and the image data. The evolution equation for updating $\vec{u}_S^t$ is defined as

$$\vec{u}_S^t = (A + \gamma \mathbf{I})^{-1} (\vec{u}_S^{t-1} - \mathbf{F}_{con}(\vec{u}_S^{t-1})) \quad (6.8)$$

This work is based on this evolution equation introduced by Kass *et al.* that includes the internal smoothness forces $((A + \gamma \mathbf{I})^{-1})$, and focuses on designing new regularizing forces for the AC model in order to segment the overlapped fibre/cell objects. Since the new AC model is developed by considering the snake as a ‘hypothetical cell’, it is called Cellsnake.

The key of Cellsnake is to use the constraint forces $\mathbf{F}_{con}(\cdot)$ derived from the cellular shape characteristics to regulate the deformation of the snakes, and these forces are referred to as the cellular forces $\mathbf{F}_{cell}$. This is motivated by the idea of using the non-overlap regions in the ROI to ‘predict’ the occluded shapes of the fibres in the overlap regions. These constraint forces are the internal forces of the snake for maintaining a cellular shape, and hence are referred to as the cellular forces $\mathbf{F}_{cell}$. Also, $\mathbf{F}_{cell}$ are combined with the two external forces $\mathbf{F}_{env}$ derived from the image data, so that the deformation of the snake is also regularized by the image information and hence achieves a better fitting result. Based on the formulation in Eq. (6.8), the conditional forces $\mathbf{F}_{con}(\cdot)$ employed by the Cellsnake is defined as

$$-\mathbf{F}_{con}(\vec{u}_S^t) = \mathbf{F}_{cell}(\vec{u}_S^t) + \mathbf{F}_{env}(\vec{u}_S^t) \quad (6.9)$$

Similar to the internal smoothness forces, $\mathbf{F}_{cell}$ are the constraint forces that arise from the internal growing energy of the AC as a ‘cell’. On the contrary, $\mathbf{F}_{env}$ are the image derived external forces that ‘help’ the cell-snake to grow in the desired region (ROI), and prohibit the further development beyond the boundary of the ROI. In this case, $\mathbf{F}_{env}$ acts like a scaffold that controls the development of the cell-snake whereas $\mathbf{F}_{cell}$ enforces the expected cell-shape properties of the snake regardless of the ROI shape. Consequently, in this work, $\mathbf{F}_{cell}$ is said to form the ‘cellular module’ of Cellsnake and $\mathbf{F}_{env}$ forms the ‘environment module’. Before showing the ‘cellular module’ which is the core of Cellsnake, the construction of the ‘environment module’ is firstly discussed in the following section.

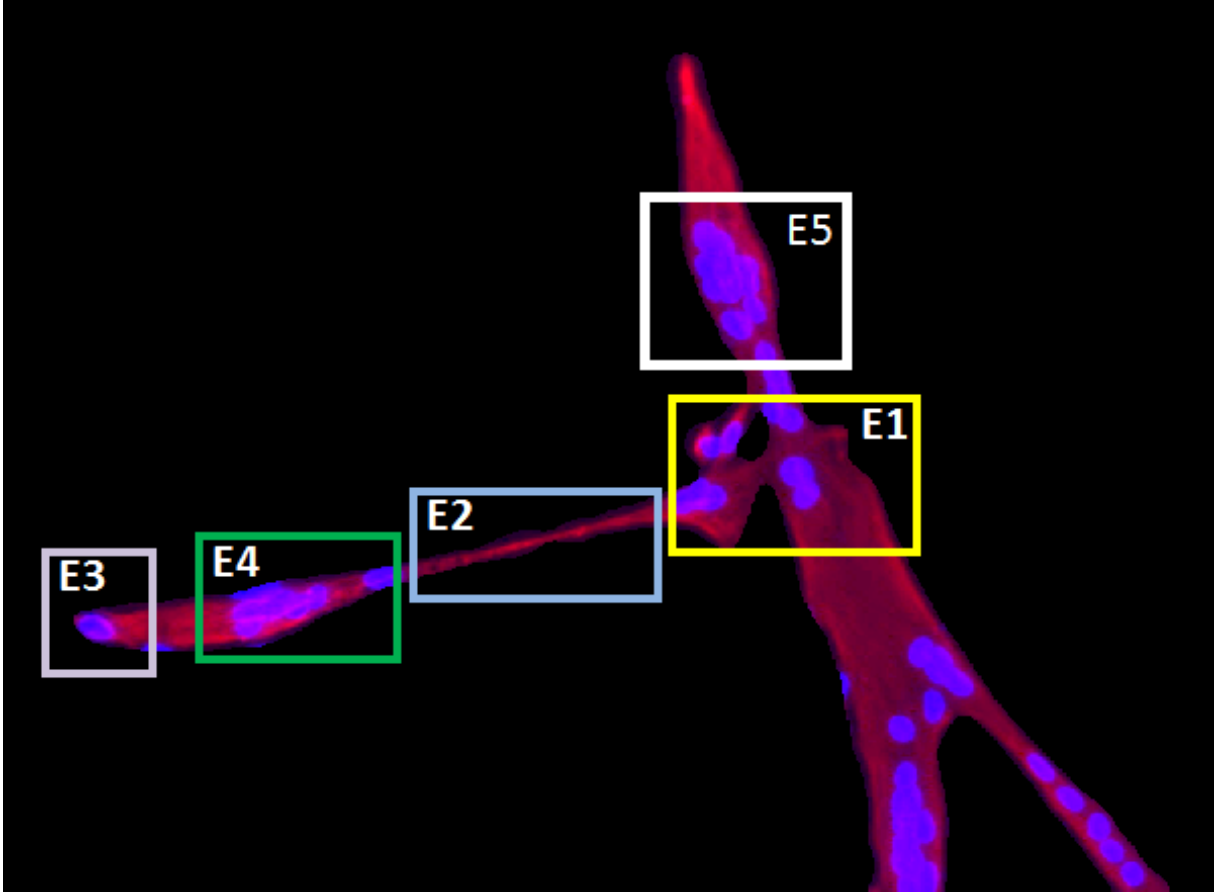


Figure 6.7: Sample of ROI with overlapped objects *This figure shows a ROI from the region ‘A’ of Fig. 6.1. The highlighted rectangular regions are the reference regions to help illustrate the ideas in later sections.*

6.4.1 Environment Forces F_{env}

The ‘environment module’ (F_{env}) is formed by two image forces, the VFC forces f_{vfc} and the background forces f_{b} , and is defined as follows

$$F_{\text{env}}(\vec{u}_S^t) = k_{\text{in}} f_{\text{vfc}}(\vec{u}_S^t) + k_{\text{out}} f_{\text{b}}(\vec{u}_S^t) \quad (6.10)$$

where k_{in} and k_{out} are the regularizing weights that control the effects of the two forces on the AC model. The image forces are derived from the geometry of the objects in the image, as shown in Fig. 6.8. The first term f_{vfc} is a modified vector field convolution (VFC) force (the ‘yellow’ arrows in Fig. 6.8) that is derived from the image topology of the cell/fibre region, i.e. the combined object or ROI. The second term f_{b} is a ‘background’ force (the ‘red’ arrows in Fig. 6.8) induced by the area outside the ROI, i.e. the background.

When considering F_{env} as a scaffold, f_{b} acts like the container that prevents the snakes growing outside the ROI, and f_{vfc} stimulates the growth of the snake towards the ‘internal

boundaries' between the cell candidates and the boundary of the ROI.

$\mathbf{f}_{\text{vfc}}$ is based on the VFC force introduced in by Li and Acton [58]. However, due to the nature of the data for this application, the VFC is modified to guide the snake from bright regions to the dimmer regions inside the ROI instead of guiding it to the regions of high gradient. The dimmer regions of the ROI are located near ROI boundary or could indicate a possible internal boundary between cells that are part of a fibre.

6.4.1.1 Vector Field Convolution Force $\mathbf{f}_{\text{vfc}}$

Although the shape information of each individual 'cell candidate' is very difficult to define from the intensity profile of 'parent' fibres, the image information of the cell candidates is still retained in the intensity pattern within the ROI corresponding to a fibre. A hypothesis is made that the image topology of the ROI can somehow 'guide' the evolution of the snake toward to the 'real' shape of the cell candidate. The reason for this hypothesis is that when the snake is initially assigned by the contour of a nuclear region, the trend of the evolution is from the high intensity region (the nuclear region) to the low intensity region (the boundary), and the dimmer regions in the ROI are usually where the boundaries of the cell candidates are located. This implies that the reverse propagation against the trend, i.e. from dimmer region to brighter region, is restricted as the snake may be propagating over the boundary of the corresponding 'cell candidate' to another 'cell candidate'. This can be modelled as the contour following a flow from the nuclei to the boundary.

In order to create these 'flows' in the whole area of the object, the modified VFC technique is applied. The vector field kernel consists of weighted vectors that point outward from the centre of the kernel whose magnitude decreases as the distance from the centre increases, and is expressed as

$$\vec{k}(\vec{x}) = m(\vec{x})\vec{n}(\vec{x}) \quad (6.11)$$

where $k(\cdot)$ is a vector field kernel with magnitude $m(\cdot)$ and direction $n(\cdot)$ at location $\vec{x}$. Note that $n(\vec{x})$ is a unit vector pointing outward from the centre of the kernel at position $\vec{x}$, and $m(\vec{x})$ is a 21×21 Gaussian kernel with $\sigma = 6$, and $\sum m(\vec{x}) = 1$. This means the deformation is fast at the beginning and slows down when the AC reaches the boundary.

The modified VFC in this work induces forces that push the AC from bright to darker regions inside the ROI and is defined as follows

$$\mathbf{f}_{\text{vfc}}(\vec{x}) = \frac{I(\vec{x})}{255} \sum_{\vec{y}_x \in N_x} (I(\vec{x}) - I(\vec{y}_x)) \vec{k}(\vec{y}_x) \quad (6.12)$$

where $\vec{y}_x$ is the pixel inside the neighbourhood window N_x of $\vec{x}$. Since the 'cell candidates' are within the ROI, the guiding forces $\mathbf{f}_{\text{vfc}}$ only to occur inside the ROI which is shown by the 'yellow' arrows in Fig. 6.8.

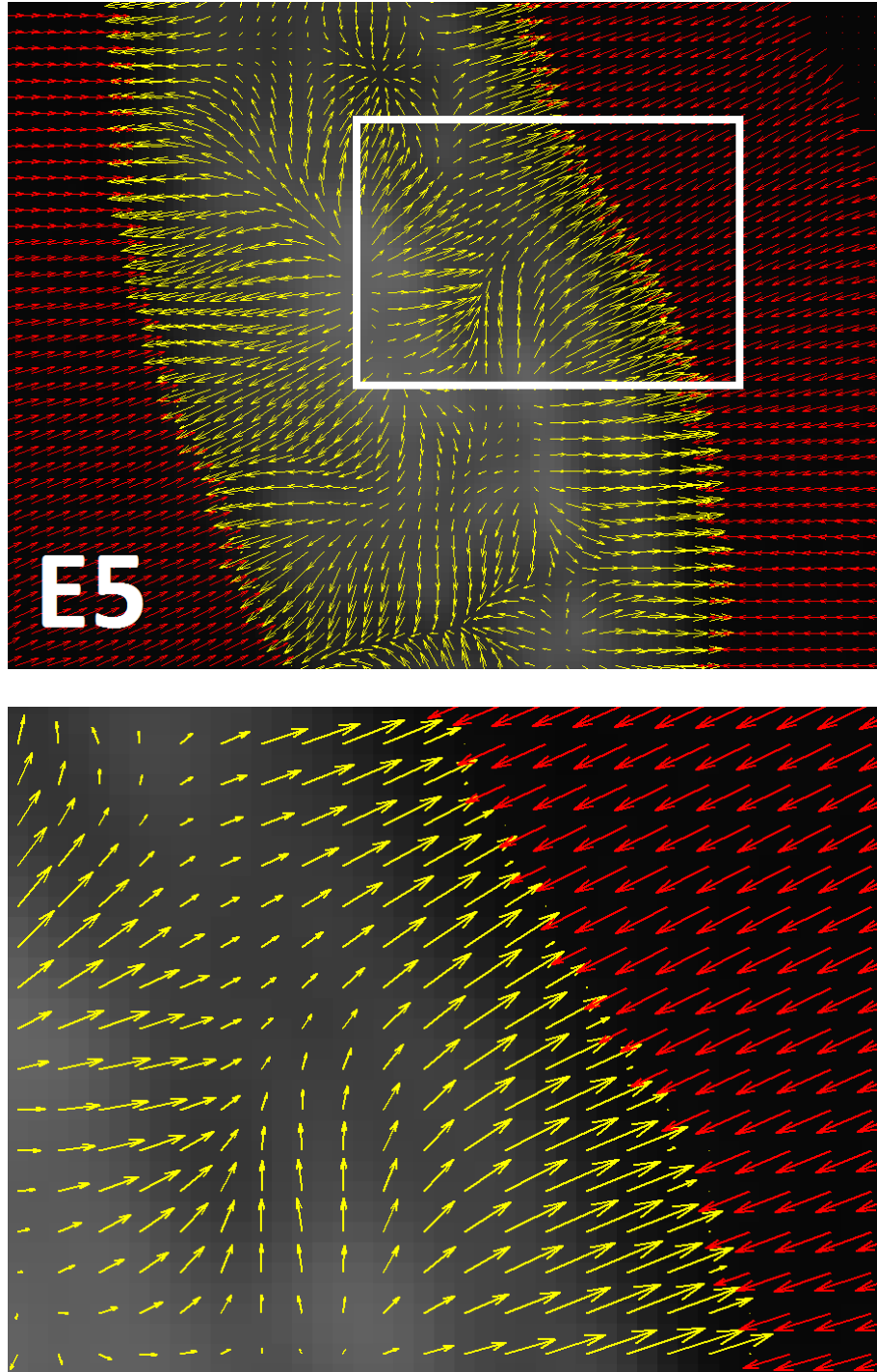


Figure 6.8: Sample of the VFC force field. The top image displays the force field of ‘growth environment’ in region ‘E5’ of Fig. 6.1, where the yellow arrows are $\mathbf{f}_{\text{vfc}}$ and the red arrows are $\mathbf{f}_{\text{b}}$. The bottom image is a zoom of the white region in the top image.

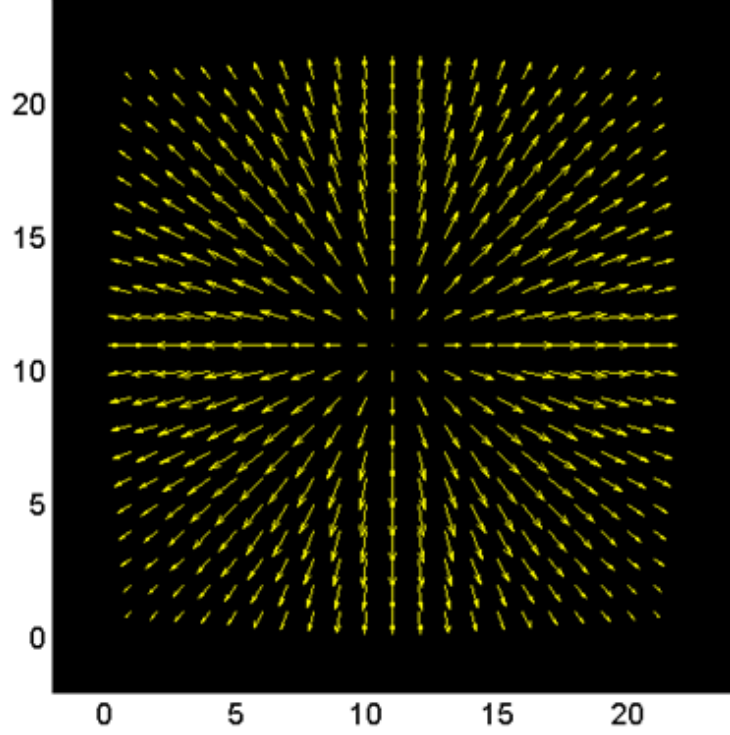


Figure 6.9: The VFC kernel (*This kernel is generated by the weighted vector that points outward from the centre of the kernel.*)

6.4.1.2 Background Force $\mathbf{f}_b$

The area that is outside of the ROI is referred to as the background of the ROI and the background forces $\mathbf{f}_b$ are derived from this area. Since the modified VFC forces $\mathbf{f}_{\text{vfc}}$ only push the ACs toward to the boundaries and are null in the background, the purpose of introducing $\mathbf{f}_b$ is to prevent the snake propagating outside the ROI ('leakage') when the target fibres/cells are within the given ROI. Leakage can occur when the snake is driven by the 'cellular module' which constraints the evolution regardless of the image topology. In order to prevent this leakage, a vector field is constructed in the background area that 'pushes' the 'leaking' contours back to the ROI, and the forces induced by the vector field are hence called the background forces $\mathbf{f}_b$.

As there is not much texture information rather than 'black' in the background, the vector field is simply constructed based on the binary image $\mathbf{M}_B$ of the background where the background region set to '1' and the ROI set to '0'. The directions and 'power' of the vector field are defined by the gradient of a blurred version of the binary image (Fig. 6.10) that is filtered by a Gaussian kernel. Accordingly, the formulation of $\mathbf{f}_b$ is expressed as

$$\mathbf{f}_b(\vec{x}) = \begin{cases} |(G * \mathbf{M}_B)(\vec{x})| \vec{v}_B(\vec{x}) & \text{if } \nabla \mathbf{M}_B \neq 0 \\ 0 & \text{if } \nabla \mathbf{M}_B = 0 \end{cases} \quad (6.13)$$

where G is a 7×7 Gaussian filter with $\sigma = 1.5$, and $\vec{v}_B(\vec{x})$ is the unit vector of the gradient



Figure 6.10: The blurred background mask. (*This mask is originally a binary mask for the area outside the ROI, i.e. the background. The background is therefore derived by the gradient of the blurred version of the background mask.*)

$\vec{\nabla}(G * M_B)(\vec{x})$ at pixel site $\vec{x}$. The background binary image is obtained by thresholding the object image $\mathbf{I}_{ROI}$ with a user defined threshold T_m as $\mathbf{M}_B = \mathbf{I}_{ROI} < T_m$ (where $T_m = 40$ here). The resultant background vector field is shown by the ‘red arrows’ in Fig. 6.8. The background forces $\mathbf{f}_b$ only take place in the background region, and is combined with $\mathbf{f}_{vfc}$ to form a complete environment force field $\mathbf{F}_{env}$ that constrains the evolution of the snake inside and outside the ROI.

6.4.2 Cellular Module $\mathbf{F}_{cell}$

Recall that $\mathbf{F}_{cell}$ is designed from the shape pattern of the single cells apparent in the images, and enable the snake to behave like a hypothetical cell. The cellular shape influences are summarized by some combination $H(\cdot)$ of four cellular forces as follows

$$F_{cell}(\vec{\mathbf{u}}_S^t) = H(f_e(\vec{\mathbf{u}}_S^t), f_\kappa(\vec{\mathbf{u}}_S^t), f_m(\vec{\mathbf{u}}_S^t), f_{cs}(\vec{\mathbf{u}}_S^t)). \quad (6.14)$$

Here, $\mathbf{f}_e$ is an elliptically weighted force that enables the snake to develop to an elongated shape. The weights are large when the forces are parallel to the major axis, and are small when the forces pointing at the perpendicular axis (the minor axis). While $\mathbf{f}_e$ is the major growing force, $\mathbf{f}_\kappa$ is an auxiliary growing force that uses the curvature of the AC to aid the development of the snake over the narrow regions in the ROI which is restricted by the smoothing effects of the internal smoothness forces in Eq. (6.8). $\mathbf{f}_m$ is a momentum force that regulates the rate of evolution of the snake with respect to time (iterations t). This is because the ‘motion’ of the cell membrane during the development is assumed to be ‘smooth’, and so the rate of evolution

of the AC at each snaxel should also be ‘smooth’. The last cellular force $\mathbf{f}_{cs}$ is a centre shift force that prohibits the propagation of the snake at one direction when the propagation at the opposite direction has been ‘stopped’ by the boundary of the ROI. This force is based on the assumption that the nuclear region is always located at or close to the centre of the cell body.

These four forces form the basis of the ‘cellular module’. Each of these forces will be discussed in turn. Subsequently, some modifications to the elliptical force are described. These modifications were applied to address some practical limitations of the technique uncovered during testing.

6.4.2.1 Elliptical Force $\mathbf{f}_e$

Again, as shown in Fig. 6.1, the shapes of the skeletal cells tend to be elongated [40]. This means the development of a skeletal cell is more facilitated in one direction than the perpendicular direction. Also, since the membranes of the single cells tend to be smooth and curved, an assumption is made that the shape of a cell can be approximated by an ellipse. Consequently, the snake that fits the cell is constrained to be elliptical, and the propagation in the major axis direction is faster than the minor axis direction.

This constraint gives rise to elliptical growing forces $\mathbf{f}_e$ that encourage the snake to grow in an elongated manner and hence facilitate the development in the major axis direction. The key of $\mathbf{f}_e$ is to assign elliptical magnitudes $\mathbf{w}_\theta$ to the unit growing vectors $\vec{\mathbf{v}}_e$ (the directions of $\mathbf{f}_e$) at the snaxels, where $\mathbf{w}_\theta$ become larger when $\vec{\mathbf{v}}_e$ are parallel to the major axis. The elliptical forces $\mathbf{f}_e$ are similar to the pressure force of the ‘Balloon’ technique introduced by Cohen in [25], but in this case $\mathbf{f}_e$ is directionally dependent.

Also, the elliptical constraint is similar to active shape model (ASM). The difference is that ASM is using a statistical model trained from a set of reference samples with very similar objects, but in this case, the elliptical constraint is estimated from the current shape of AC at each iteration. Although the single cells are simpler than the fibres in the images and have the similarity that they tend to be elongated, the variance of the cellular shapes is still large. Hence, the elliptical constraint estimated from the AC at each iteration is preferable, as this can control the deformation of the AC to be elongated but not force it to become an ellipse.

As shown in Fig. 6.11, in order to set the values of $\mathbf{w}_\theta$ (the length of the ‘blue arrows’), the first step is to fit the AC with an ellipse and determine the major axis ($\vec{\mathbf{d}}_{E1} = (h_{E1}, k_{E1})$) and the minor axis $\vec{\mathbf{d}}_{E2} = (h_{E2}, k_{E2})$ of the snake, and the centre ($\vec{\mathbf{c}}_E$) of the ellipse.

The second step is to setup the growing vector $\vec{\mathbf{v}}_e$ (direction of $\mathbf{f}_e$). As illustrated in Fig. 6.11, $\vec{\mathbf{v}}_e$ at each snaxel is defined as

$$\vec{\mathbf{v}}_e(s, t) = \lambda \vec{\mathbf{v}}_c(s, t) + (1 - \lambda) \vec{\mathbf{n}}_\phi(s, t) \quad (6.15)$$

where λ is a adjustable weight, $\vec{\mathbf{n}}_\phi(s, t)$ is the unit normal vector of the AC at snaxel s , $\vec{\mathbf{v}}_c(s, t)$

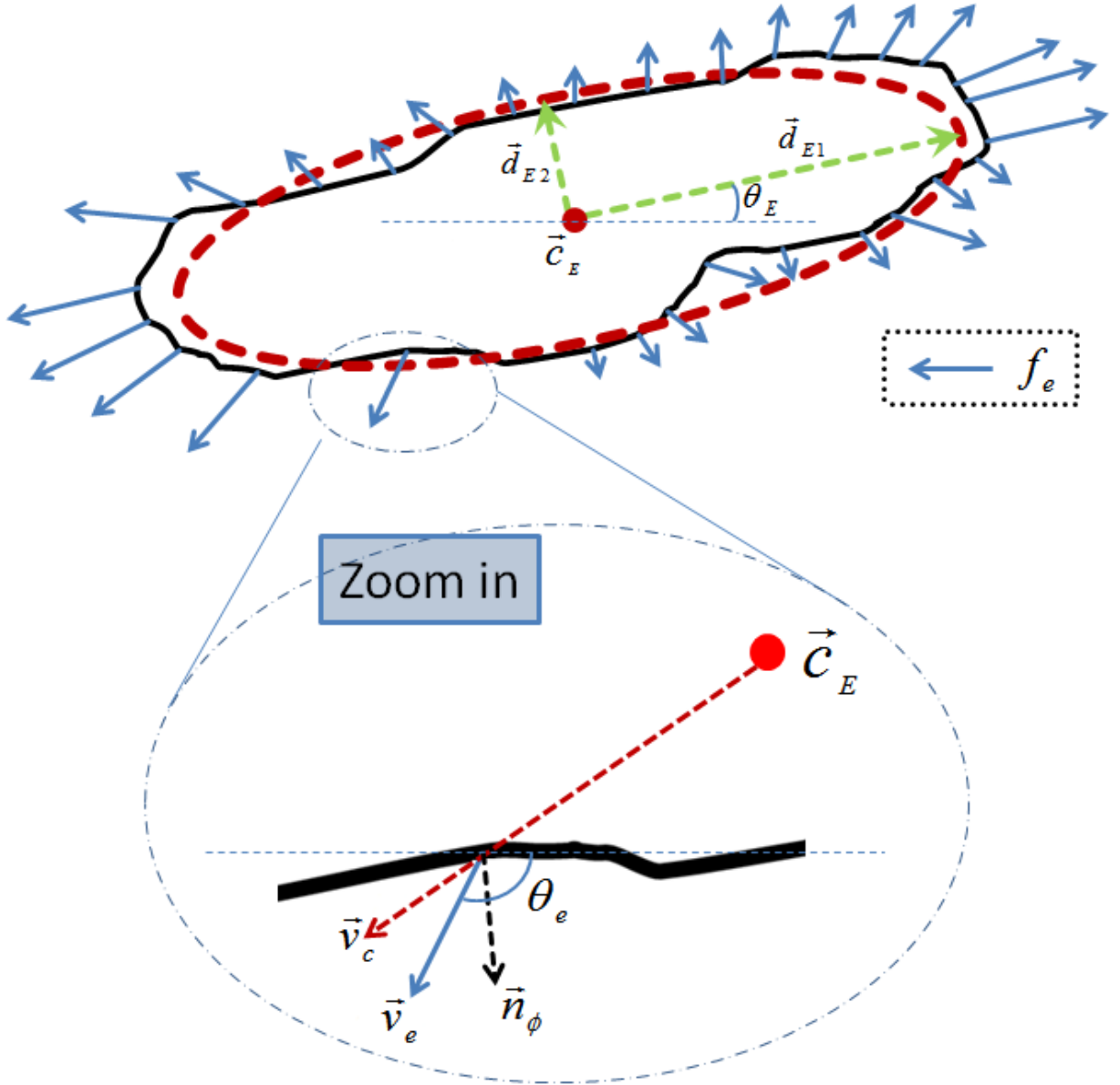


Figure 6.11: Illustration of the idea of elliptical force $\mathbf{f}_e$ (The snake is fitted with an ellipse at each iteration. The fitted ellipse has the centre at $\vec{c}_E$, the major axis $\vec{d}_{E1}$, the minor axis $\vec{d}_{E2}$, and orientation θ_E which is also the orientation of $\vec{d}_{E1}$. The direction of the elliptical force $\mathbf{f}_e$ ('blue arrows') at each snaxel is determined by the normal direction of the contour at that snaxel and its orientation with respect to $\vec{c}_E$. $\vec{n}_\phi$ is the unit normal vector of the contour at each snaxel. $\vec{v}_c$ is the unit vector from $\vec{c}_E$ to each snaxel and with orientation θ_e . $\vec{v}_e$ is a unit vector that indicates the direction of the elliptical force $\mathbf{f}_e$ and is set by combining $\vec{n}_\phi$ and $\vec{v}_c$.)

is the unit vector in the direction of s with respect to $\vec{c}_E$ that expressed as

$$\vec{v}_c(s_n, t) = \frac{\vec{u}_s^t - \vec{c}_E}{|\vec{u}_s^t - \vec{c}_E|} \quad (6.16)$$

The normal vectors $\vec{n}_\phi$ indicate a potential deformation of the AC that extends the shape from previous iteration. In contrast, $\vec{v}_c$ represents an elliptical deformation which is typically independent of the actual shape of the snake. Hence, λ is used to control how much the snake is growing in an elliptical manner (the ‘red’ shape in Fig. 6.11), where a larger λ forces the snake to deform like an ellipse, or simply ‘extends’ the AC (the ‘black’ shape in Fig. 6.11) in the normal directions when $\lambda = 0$.

The third step is to compute w_θ based on the correlation between the growing directions $\vec{v}_e$ of the snaxel and the direction of the major axis $\vec{d}_{E1}$. w_θ are large when $\vec{v}_e(\cdot)$ is in the same direction or the opposite direction as $\vec{d}_{E1}$. Conversely, w_θ are small, when $\vec{v}_e(\cdot)$ is perpendicular to $\vec{d}_{E1}$. Also, w_θ are set in the range of $[|\vec{d}_{E2}(t)|/|\vec{d}_{E1}(t)|, 1]$. This is because when the snake is small at the beginning ($|\vec{d}_{E2}(t)| \approx |\vec{d}_{E1}(t)|$), any directions can be the ‘actual’ evolution direction of the snake and hence the growing directions of the snaxels are not biased. But when the snake becomes bigger ($|\vec{d}_{E2}(t)| < |\vec{d}_{E1}(t)|$), the ‘actual’ evolution direction $\vec{d}_{E1}(t)$ would become more confident and the deformation along $\vec{d}_{E2}(t)$ would be prohibited more and more. Consequently, w_θ are estimated by

$$w_\theta(s, t) = \left| \frac{\vec{d}_{E1}(t)}{|\vec{d}_{E1}(t)|} \cdot \vec{v}_e(s, t) \right| \frac{|\vec{d}_{E1}(t)| - |\vec{d}_{E2}(t)|}{|\vec{d}_{E1}(t)|} + \frac{|\vec{d}_{E2}(t)|}{|\vec{d}_{E1}(t)|} \quad (6.17)$$

Since the magnitudes w_θ and the directions $\vec{v}_e(\cdot)$ are defined, the elliptical forces $\mathbf{f}_e$ are simply expressed as

$$\mathbf{f}_e(\vec{u}_s^t) = w_\theta(s, t) \vec{v}_e(s, t) \quad (6.18)$$

6.4.2.2 Curvature Force $\mathbf{f}_\kappa$

The purpose of the curvature force $\mathbf{f}_\kappa$ is to aid the snake to propagate over very narrow regions like the ‘E2’ region in Fig. 6.7. When a snake reaches a narrow region, the snaxels near the boundary of the ROI are attracted to the boundary by $\mathbf{F}_{\text{env}}$. This ‘boundary attraction’ cancels the growing forces that push the snaxels along the narrow region, and only small amount of snaxels that move approximately horizontally to the region boundary have enough ‘energy’ to overcome the attraction and remain ‘active’. Because of this, there can be insufficient active snaxels to overcome the internal smoothness force (Eq. (6.2)). Hence, the growth of the snake into the narrow region is severely restricted and becomes very slow and can even stop.

Since the goal of the snake is to fit a hypothetical cell candidate in the ROI, the deformation being terminated in the middle of a narrow region is not acceptable when the ‘cell’ is likely to cover the whole region. A solution is found from the ‘arc’ of the AC in the narrow region, which is sharper in narrower regions. Based on this observation, a curvature force $\mathbf{f}_\kappa$ is introduced to the AC model to enhance the growing ‘energy’ of the snaxels with high curvature.

Similar to the applications of curvature in [22, 39], the curvature force $\mathbf{f}_\kappa$ for each snaxel is defined as

$$\mathbf{f}_\kappa(\vec{u}_s^t) = \kappa(s, t) \vec{n}_\phi(s, t) \quad (6.19)$$

where $\kappa(s)$ is the curvature of the AC at snaxel s at iteration t . Suppose the curve of the snake is represented by the function $\vec{u}(s) = (h(s), k(s))$, $\kappa(s)$ is formulated as

$$\begin{aligned} \kappa(s) &= \left| \frac{\frac{\partial^2 u}{\partial s^2}}{\left(1 + \left(\frac{\partial u}{\partial s}\right)^2\right)^{1.5}} \right| \\ &= \left| \frac{\frac{\partial h}{\partial s} \frac{\partial^2 k}{\partial s^2} - \frac{\partial k}{\partial s} \frac{\partial^2 h}{\partial s^2}}{\left(\left(\frac{\partial h}{\partial s}\right)^2 + \left(\frac{\partial k}{\partial s}\right)^2\right)^{1.5}} \right| \end{aligned} \quad (6.20)$$

In addition, $\mathbf{f}_\kappa$ also aids the snake to grow in an elongated shape. This is because when $\mathbf{f}_e$ regulates the snake to deform as an ellipse, the ‘arcs’ of the AC along the major axis would be sharper than the ‘curves’ along the minor axis. This means the ‘arcs’ in the elongated directions gain larger curvature forces and hence the snake becomes further elongated.

6.4.2.3 Momentum Force $\mathbf{f}_m$

The convergence of the snake at the object boundary is an important consideration in designing the AC models. Convergence problems can be due to the interaction of forces when the snake reaches the boundary, causing the snake to oscillate across the ROI boundary. The oscillation becomes ‘larger’ when the step size γ in Eq. (6.6) increases. Since the oscillation is in fact a series of sudden reverse movements, a force that is based on the movement difference at each snaxel between two iterations is added to the model in order to damp the growth of the snake.

Here, an assumption is made that when a snaxel is continuously moving in a certain direction, the snaxel will develop a momentum that influences the next move to remain in that direction. This momentum effect is represented by the momentum forces $\mathbf{f}_m$ in the AC model. In this work, the momentum force $\mathbf{f}_m$ at each snaxel is simply derived from the dot product between the moving directions of the snaxel at the present iteration (t) and the previous iteration ($t-1$), which is defined as

$$\mathbf{f}_m(\vec{u}_s^t) = \left(\vec{v}_e(s, t) \cdot \left(\frac{\vec{u}_s^t - \vec{u}_s^{t-1}}{|\vec{u}_s^t - \vec{u}_s^{t-1}|} \right) \right) \vec{v}_e(s, t) \quad (6.21)$$

Since $\mathbf{f}_e$ is the major ‘growing’ force and dominates the deformation of the snake, $\vec{v}_e$ (direction of $\mathbf{f}_e$) is chosen as the present moving direction of each snaxel. When $\vec{v}_e$ is towards the previous moving direction, the dot product will be positive and the snaxels will move faster in that direction. Otherwise, when the snaxel is moving in the opposite direction, the dot product becomes negative, and the reverse movement is restricted. Also, $\mathbf{f}_e$ helps the snake to deform faster when growing directions remain approximately the same, and change the shape smoothly when the growing directions are changed.

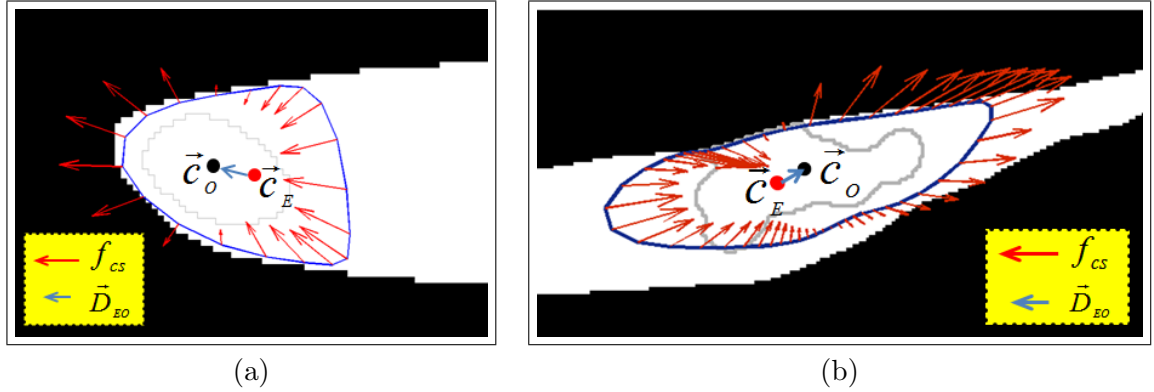


Figure 6.12: Illustrations of the centre-shift forces $\mathbf{f}_{cs}$. This figures shows that the deformation of the ACs (blue curve) is constrained by $\mathbf{f}_{cs}$ (red arrows) derived from shifting the centre $\vec{c}_E$ (of fitted ellipse) of the snake to the centre $\vec{c}_O$ of nuclear region (grey contour). The displacement from $\vec{c}_E$ to $\vec{c}_O$ is denoted as $\vec{D}_{EO}$. (a) and (b) are respectively related to the ‘E3’ and ‘E4’ regions in Fig. 6.7

6.4.2.4 Centre Shift Force $\mathbf{f}_{cs}$

From observation, the nuclear region of single cells (the ‘yellow’ objects in Fig. 6.3) is roughly near the centroid of the cell shape. However, the forces described in the previous sections cannot enforce this constraint on the snake and so it is possible that the nucleus is displaced from the centre of the snake (Fig. 6.12(a)). This can affect the segmentation of cells that ‘adjoin’ (but not overlap) to larger neighbouring cells/fibres in the same ROI, as the displacement of the snake centroid would cause the snake to erroneously evolve into the neighbouring cell/fibre region. This could potentially cause the cell to be estimated as part of the neighbouring cell/fibre during the merge process.

In order to solve this problem, a cellular force is introduced to maintain the centre of the snake close to the centre of the associated nuclear region, and is called the centre shift force $\mathbf{f}_{cs}$. The idea of $\mathbf{f}_{cs}$ is to push the snaxels towards or away from the centre $\vec{c}_O$ of the nuclear region, so that the centre $\vec{c}_E$ of the fitted ellipse is moved toward $\vec{c}_O$.

The direction and magnitude of $\mathbf{f}_{cs}$ at each snaxel depends on the direction of the snaxel with respect to $\vec{c}_O$ and the displacement $\vec{D}_{EO}$ from $\vec{c}_E$ to $\vec{c}_O$. The direction from snaxel s to $\vec{c}_O$ is represented by a unit vector $\vec{v}_{cs}(s, t)$ and defined as

$$\vec{v}_{cs}(s, t) = \frac{\vec{u}_s^t - \vec{c}_O}{|\vec{u}_s^t - \vec{c}_O|} \quad (6.22)$$

The centre shift forces $\mathbf{f}_{cs}$ should counteract the growing forces (mainly $\mathbf{f}_e$) when the relative orientation ($\vec{v}_{cs}$) of the snaxel with respect to the snake centre is opposite to the orientation of $\vec{D}_{EO}$. On the other hand, $\mathbf{f}_{cs}$ should enhance the growing forces when $\vec{v}_{cs}$ and $\vec{D}_{EO}$ have similar

orientations. Therefore, $\vec{v}_{cs}$ is defined as

$$f_{cs}(\vec{u}_s^t) = \left(\left(\frac{\vec{D}_{EO}(t)}{\bar{D}_{uO}} \right) \cdot \vec{v}_{cs}(s, t) \right) \vec{v}_{cs}(s, t) \quad (6.23)$$

where $\bar{D}_{uO}$ is the mean distance between $\vec{c}_O$ and all the snaxels. As shown in Fig. 6.12(b), $\mathbf{f}_{cs}$ not only constrains the evolution of the snakes at the ‘corners’ of the ROI, but also help the snake to propagate in the narrow regions. This occurs because when the snake develops faster on one side at the wider region, the centre of the snake $\vec{c}_E$ would be ‘moved’ away from $\vec{c}_O$, and this triggers $\mathbf{f}_{cs}$ to facilitate the development on the other side at the narrow region.

6.4.3 Practical Modifications

So far, the four components of the cellular module have been defined in the previous sections. As mentioned previously, these constraint forces enforce the snake to evolve in a cellular pattern and the hence estimate the shapes of the overlapped fibre/cell obejects in the ROI. However, due to the complexity of the object shapes in this case and computational limitation, the new AC model encountered several practical problems in shape estimation. To solve these problems, two modifications are applied to the estimation of the elliptical force $\mathbf{f}_e$ (the major growing force) in order to improve the estimation result of the model. These modifications are the rotational moment and the force deflections. The rotational moment is proposed to adjust $\mathbf{f}_e$ in order to prevent the leakage of the snake when there is no snaxel in the leaking region to take the effect of the background force $\mathbf{f}_b$. On the other hand, the force deflections are applied to adjust the orientation of $\mathbf{f}_e$, and consist of the repulsive deflection and the pulling deflection. The purpose of the repulsive deflection is to further enhance the growth of the snake in the narrow regions, and the pulling deflection reduces the divergence of the deformation in the occluded regions.

6.4.3.1 Rotational Moment

In some particular cases (Fig. 6.13), the AC may leak slightly into the background area around an overlap region. Normally, the background forces $\mathbf{f}_b$ would prevent this, but in practice due to the finite number of snaxels used to represent the AC for computational reasons, there may be no snaxel in this part of the contour. When this happens, even though $\mathbf{f}_b$ provides the desired correction, there is no snaxel to respond to it.

To solve this problem the existing snaxels which may be elsewhere need to be moved in such a way as to respond to this ‘leakage’. In fact, from observation, the leakage region can give a hint of the deformation that would give a better overall segmentation result. This is illustrated in Fig. 6.13(a) that the AC is slightly rotated in anti-clockwise about $\vec{c}_E$ by a force in the ‘downward leaking’ region (marked by the yellow dot). This rotation can help the AC stay inside the ROI and and help to keep the major axis $\vec{d}_{E1}(t)$ pointing towards the portion of occluded object on the other side of the occluded region. Otherwise, the development of the

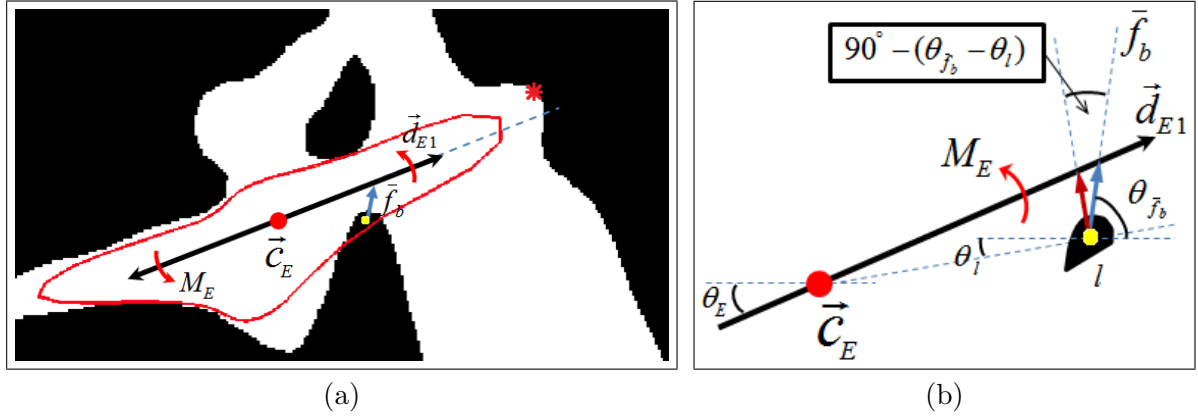


Figure 6.13: Illustrations of rotational moment M_E . (a) is related to the ‘E1’ region in Fig. 6.7 and shows the idea of rotational moment caused by the ‘leakage’ regions. This moment is induced by the mean $\bar{f}_b$ of background forces (at the ‘yellow’ centre) in the leakage region and acts on the AC to rotate the fitted ellipse (change the orientation θ_E of the major axis $\vec{d}_{E1}$), like a lever fixed at the centre $\vec{c}_E$, and hence alter the elliptical force $\bar{f}_e$ in order to give a better estimation (‘red *’). (b) shows the mechanism of estimating M_E from $\bar{f}_b$.

AC could stop at the boundary of the occluding object and gives an inappropriate segmentation result.

This hypothetical rotational effect to the AC motivates the proposal of the ‘rotational moment’ (M_E) in this work. Suppose the leakage region is denoted as l (group of pixels), the ‘upward lifting’ force is set by the mean ($\bar{f}_b(l)$) of the background forces in L , since $\bar{f}_b$ pushes the leaking AC back to the ROI. The idea of the rotational moment (M_E) is to change the growing forces at the snaxels so that the resultant deformation is equivalent to rotating the AC about $\vec{c}_E$ with the force $\bar{f}_b(l)$. Since the major growing forces ($\bar{f}_e$) depend on the orientation θ_E of the major axis $\vec{d}_{E1}$ of the fitted ellipse, the ‘rotational effect’ on the growing forces is equivalent to altering θ_E according to the value of M_E . As shown in Fig. 6.13(b), the ‘rotational moment’ M_E is defined as an angular moment in terms of the perpendicular component of $\bar{f}_b(l)$ ‘acting on’ $\vec{d}_{E1}$ and is expressed as

$$M_E(l) = \bar{f}_b \left| \sin \left(\theta_{\bar{f}_b}(l) - \theta_E \right) \right| \quad (6.24)$$

where $\theta_{\bar{f}_b}(l)$ is the orientation angle of $\bar{f}_b(l)$. θ_l is the orientation of the centre of l with respect to $\vec{c}_E$.

Suppose a snake has L leakage regions, the resultant orientation $\theta_E^*(t)$ of the snake after influencing $\theta_E(t)$ with the total rotational moment is defined as

$$\theta_E^* = \theta_E + \sum_{l=1}^L \frac{A(l)}{a_M} M_E(l) \quad (6.25)$$

where $A(l)$ is the size of the leakage region l , and a_M is a regularizing factor. Note that $\frac{A(l)}{a_M}$

controls the rotating effect of $\mathbf{M}_E$ on the AC depending on the size of the leakage region. The purpose of this control is to increase the rotational effect when the leakage area is large, and hence ‘guide’ the AC back into the ROI relatively quicker. Also, when the AC has multiple leakage regions that provide ‘opposite’ rational moments, the region with larger size has higher priority and hence has larger rational moment on the AC. The same rotation is also applied to the minor axis ($\vec{d}_{E2}$).

6.4.3.2 Force Deflection

Recall that the propagation of the snake can slow down or even stop in the narrow regions. There are two approaches to ease the problem. The first approach is to enhance the growing of the ‘active’ snaxels by applying the curvature force ($\mathbf{f}_\kappa$). However, in practice, due to the limited number of snaxels, this growth enhancement may not be enough to help the snake to propagate when the number of ‘active’ snaxels is too small. The second approach is to ‘activate’ the ‘inactive’ snaxels, which gives rise to a ‘force deflection’ that applies to the major growing forces ($\mathbf{f}_e$).

As shown in Fig. 6.14(a), the reason of the snaxels being inactive at the ROI boundary is that the snaxels are ‘trapped’ in the oscillation across the boundary due to the interaction of the background force $\mathbf{f}_b$ and $\mathbf{f}_e$ (the main growing force). In this case, the VFC force $\mathbf{f}_{\text{vfc}}$ is too small and hence is negligible. The purpose of the repulsive deflection is to change the orientation of $\mathbf{f}_e$ so that the move the ‘trapped’ snaxel moves along the boundary during the oscillation rather than oscillates about a ‘fixed’ point of the boundary.

The repulsive force deflection can be imagined as $\mathbf{f}_e$ of the ‘trapped’ snaxels being repel by the boundary and hence the direction is turned towards to $\vec{d}_{E1}$ without altering the magnitudes $|\mathbf{f}_e|$. The deflected forces are denoted as $\bar{\mathbf{f}}_e$. As illustrated in Fig. 6.15, the repulsive deflection is derived from the angular difference θ_r between $\mathbf{f}_e$ and $\mathbf{f}_b$ ($\theta_r = 0$ when $f_e = 0$). The deflection alters the orientation θ_e of f_e by a portion λ_r of θ_r . The deflected elliptical force is denoted as $\bar{\mathbf{f}}_e$. As a result, although $\bar{\mathbf{f}}_e$ would be reduced by $\mathbf{f}_b$, the resultant force still encourages pushing of the snaxels to move along the boundary.

Besides, the repulsive deflection, a pulling deflection is also applied to $\mathbf{f}_b$ in order to reduce the unwanted deformation divergence of the snakes in the overlap regions. Recall that the overlap regions in the ROI only contain the image information of the occluding objects but not the occluded objects. As shown in Fig. 6.14(c), when a snake that belongs to an occluded object reaches an overlap region (the ‘E1’ region in Fig. 6.7), the snaxels can be confused by the topological flow ($\mathbf{f}_{\text{vfc}}$) which starts moving away from the major axis $\vec{d}_{E1}$ and being attracted by the ‘wrong’ object. This causes the deformation of the AC to diverge from the appropriate shape (the ‘red’ curve). In the case of Fig. 6.7, the development of the AC will diverge to the wider area of the ‘occluding’ object instead of converging to the small portion of the occluded object on the other side of the occluding region.

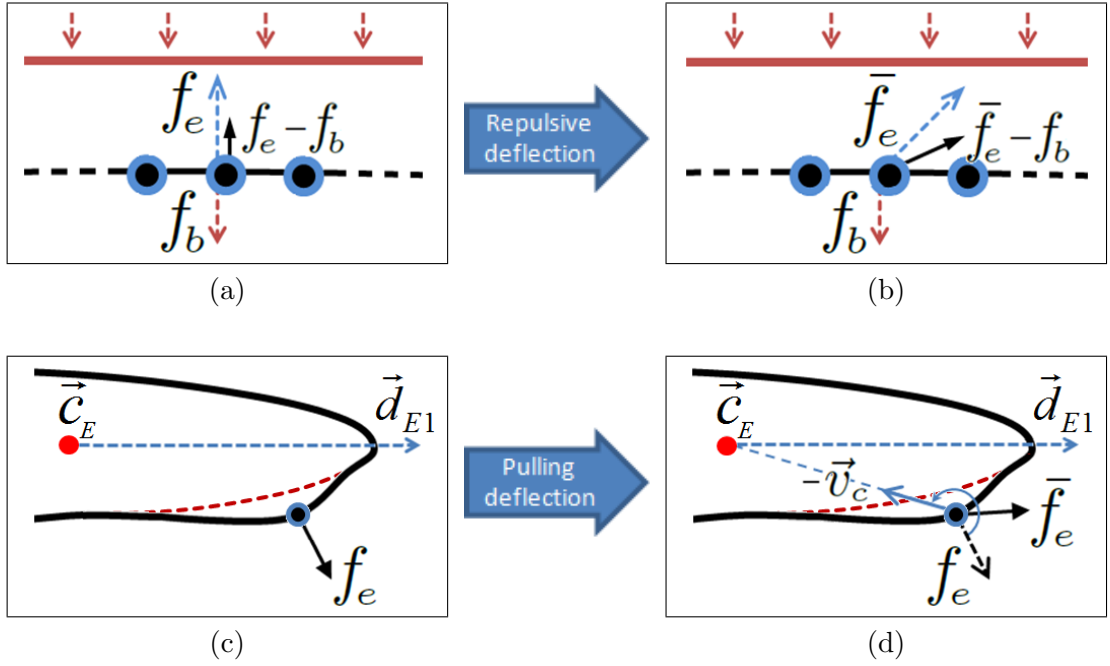


Figure 6.14: Illustrations of the repulsive and pulling force deflections applied to $\mathbf{f}_e$. (a) shows the inactive snaxels (black dots) that are ‘trapped’ at the edge of the narrow object region since there is no force to push the snaxels along the boundary (in red). (b) illustrates the idea of the repulsive force deflection being applied to the major growing forces $\mathbf{f}_e$ in order to ‘activate’ the ‘trapped’ snaxels in (a). The repulsive deflection is a hypothetical deflection from the background forces $\mathbf{f}_b$. (c) shows an unwanted deformation (divergence from the wanted deformation indicated by dash ‘red’ curve) of the snake in the overlap region (the ‘E1’ region in Fig. 6.7). This happens when the snake belongs to the occluded object and the growing directions of the snaxels are influenced by the image topology of the occluding object. In the case the growing directions are diverged away from the major axis $\vec{d}_{E1}$. (d) illustrates the idea of pulling force deflection applied to $\mathbf{f}_e$ in order to reducing the directional divergence in (c). The pulling deflection is induced by the hypothetical forces acting on the snaxels from the centre $\vec{c}_E$.

To reduce this divergence, a pulling force deflection is proposed for the growing directions of the snaxels by assuming the snaxels are ‘tied’ to the centre $\vec{c}_E$ and are pulled by $\vec{c}_E$ during the evolution in order to prevent divergence from $\vec{d}_{E1}$. As shown in Fig. 6.14(d), the pulling effect influences the snaxels to move toward $\vec{d}_{E1}$ by slightly deflecting the growing direction toward to $\vec{c}_E$ without reducing the growing power. The deflected forces $\bar{\mathbf{f}}_e$ can prevent the snaxels from moving further away from $\vec{d}_{E1}$, and hence reduce the deformation divergence. Similar to the repulsive deflection, the idea of the pulling deflection is (as illustrated in Fig. 6.15) to measure the angular difference θ_p between $\mathbf{f}_e$ and $-\vec{v}_c$ (direction from the snaxel to $\vec{c}_E$), and alter the orientation θ_e of $\mathbf{f}_e$ by a portion λ_p of θ_p to generate $\bar{\mathbf{f}}_e$. In addition, the pulling deflection can prevent diffusion at the arc of the snake when it ‘hits’ the boundary belonging to the other

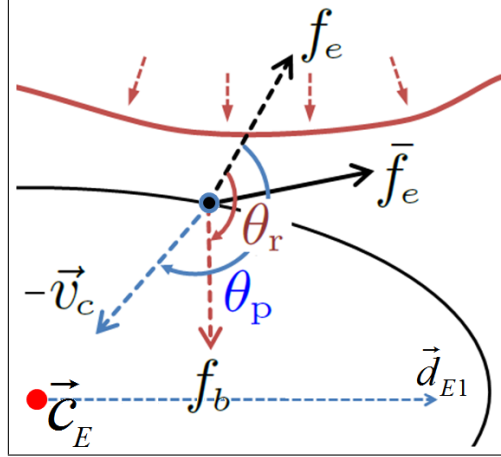


Figure 6.15: The overall force deflection to the cellular elliptical force. (c_E is the centre of the fitted ellipse. θ_r is the orientation difference between the elliptical force and the background force which induces the repulsive deflection to the elliptical force. θ_p is the orientation difference between the elliptical force and the negative orientation of the snaxel related to c_E .)

objects in the overlap regions.

Since both the repulsive and pulling deflections are applied to $\mathbf{f}_e$, the effect of the deflections are combined and regulated by λ_r and λ_p respectively. Consequently, the new orientation $\mathbf{f}_e$ is computed by

$$\bar{\theta}_e(s, t) = \theta_e(s, t) + \lambda_p \theta_p(s, t) + \lambda_r \theta_r(s, t) \quad (6.26)$$

and the deflected growing forces $\bar{\mathbf{f}}_e$ are defined as

$$\bar{\mathbf{f}}_e(\vec{u}_s^t) = |\mathbf{f}_e(s, t)| \angle \bar{\theta}_e(s, t) \quad (6.27)$$

6.4.4 Combining the Forces

Recall that the cellular module F_{cell} is a combination $H(\cdot)$ (Eq. (6.14)) of the four types of cellular forces described in the previous sections. However, the cellular forces are derived based on characteristic of the appearance shapes of the objects in the images, Cellsnake employs a number of regularizing parameters to control the effects of different constraints in the model. These parameters allow the adjustment of the effects of different constraints to suit the fibre objects with different complex shapes in different images. Also, the speed of the modelling process is based on the step size of the model, where a small step size can cause modelling process to be time consuming, and a larger step size can speed up the process but can lead to critical failure in the deformation when the snake grows too fast and the regularized constraints are not strong enough to control the model. Based on the evaluation on different settings of the forces in the model through experiments, the formulation of the model (which provides

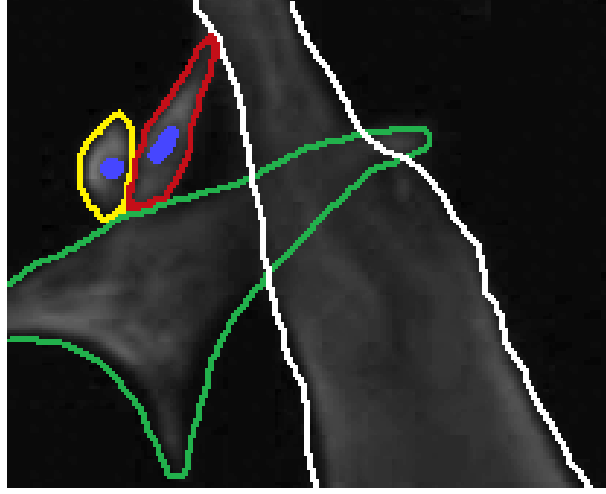


Figure 6.16: The adjoining objects in the ROI (*The objects are segmented manually by biologists. The small cells ('yellow' and 'red') with 'blue' nuclei at the centre are 'attached' to the big fibre objects highlighted in green and white.*)

reasonable results) is defined as follows

$$F_{\text{cell}}(\vec{u}_s^t) = g(\vec{u}_s^t) \left(\tau_s^t \left(w_e \bar{f}_e(\vec{u}_s^t) + w_\kappa f_\kappa(\vec{u}_s^t) + w_m f_m(\vec{u}_s^t) \right) + w_{cs} f_{cs}(\vec{u}_s^t) \right)$$

where $\{w_e, w_\kappa, w_m, w_{cs}\}$ are the regularizing factors, g and τ_s^t are the growth controlling factors. Since $\{f_e, f_\kappa, f_m, f_{cs}\}$ are derived from different aspects of the hypothetical cellular growth patterns, the 'strength' of these different cellular forces are controlled by the corresponding regularizing factors. The factors can be adjusted so that the AC model can fit the cell objects with different growth patterns.

Also, since the cellular forces are insensitive to the image topology, a Geodesic factor g is introduced to F_{cell} in order to reduce the growing 'power' (especially for large step sizes) when the snake reaches the boundary of the ROI and hence avoid the 'over' growing in the background. The idea of the Geodesic factor g is adopted from the Geodesic AC model proposed by Caselles *et al.* in [21] which is derived from the image gradient ∇I and defined as

$$g(\vec{u}_s^t) = \frac{1}{1 + |\nabla I(\vec{u}_s^t)|^2} \quad (6.28)$$

Furthermore, when the ROI consists of single cells that are 'attached' to big fibre objects in the ROI (Fig. 6.16), the deformation of the snakes associated to these cell objects can be influenced by the fibre objects (especially when the size of the cells are small). The snake associated to such a cell object can easily evolve over the 'weak boundary' by overcoming the 'weak' VFC flows (f_{vfc}) and occupy some of the area belonging to the adjoining fibre object before the centre shift force f_{cs} can counteract it. Furthermore, since the image gradient of the 'internal boundaries' in the ROI is weak, g has little response to the 'internal boundaries' to reduce the growing power.

In fact, these single cells usually have the nucleus located at the centre of the actual shape. Hence, in order to prevent the internal leakage, the snakes that are associated to these cell objects should always keep the corresponding nuclear regions at the centre of the ACs, and $\mathbf{f}_{cs}$ needs to be enhanced. However, the increasing of w_{cs} will influence the fitting performance for the large fibre objects where the nuclear region can be located a bit further from the centre. This means that the centre shift constraint is more strict for cell objects which are usually small and is more loose for fibre objects which are relatively much larger. Intuitively, since each snake is initialized at the boundaries of the nuclear regions and becomes larger during the deformation, a growth control factor τ_S^t is introduced as a weight for the other forces (i.e. not $\mathbf{f}_{cs}$) of the cellular module. The value of τ_S varies depending on the size A_{ac} of the snake (the total number of pixels inside the snake) and defined as

$$\tau_S = \begin{cases} \exp\left(-a_\tau \frac{A_{ac} - \bar{A}_{ROI}}{\bar{A}_{ROI}}\right) & , \quad A_{ac} < \bar{A}_{ROI} \\ 1 & , \quad A_{ac} \geq \bar{A}_{ROI} \end{cases}$$

where $\bar{A}_{ROI} = \frac{A_{ROI}}{N_{nuclei}}$ is assumed to be the average size of the ‘cell candidates’ in the ROI which has total size A_{ROI} and contains N_{nuclei} candidates.

When the snake starts from the nuclear region with a small size, τ_S^t is small and reduces the growing power so that $\mathbf{f}_{cs}$ becomes dominant in the cellular module. Conversely, when the snake grows larger, the centre shift constraint becomes less effective. The regularizing factor a_τ is used to control how quickly the effect of $\mathbf{f}_{cs}$ is reduced.

6.4.5 Cellsnake and the Merge scheme

The new AC model combines the ‘cellular module’ ($\mathbf{F}_{cell}$) and the ‘environment module’ ($\mathbf{F}_{env}$) by substituting Eq. (6.28), Eq. (6.10) and Eq. (6.9) to Eq. (6.8) as follows

$$\begin{aligned} \vec{\mathbf{u}}_S^{t+1} &= (\mathbf{A} + \gamma \mathbf{I})^{-1} (\vec{\mathbf{u}}_S^t - \mathbf{F}_{con}(\vec{\mathbf{u}}_S^t)) \\ &= (\mathbf{A} + \gamma \mathbf{I})^{-1} \left(\vec{\mathbf{u}}_S^t + \left(F_{cell}(\vec{\mathbf{u}}_S^t) + F_{env}(\vec{\mathbf{u}}_S^t) \right) \right) \\ &= (\mathbf{A} + \gamma \mathbf{I})^{-1} \left(\vec{\mathbf{u}}_S^t + \left(g(\vec{\mathbf{u}}_S^t) \left(\tau_S^t \left(w_e \bar{f}_e(\vec{\mathbf{u}}_S^t) + w_\kappa f_\kappa(\vec{\mathbf{u}}_S^t) + w_m f_m(\vec{\mathbf{u}}_S^t) \right) + w_{cs} f_{cs}(\vec{\mathbf{u}}_S^t) \right) \right. \right. \\ &\quad \left. \left. + (k_{in} f_{vfc}(\vec{\mathbf{u}}_S^t) + k_{out} f_b(\vec{\mathbf{u}}_S^t)) \right) \right) \end{aligned} \quad (6.29)$$

The deformation of each snake is halted when the movements of all the snaxels become very small and continue for a particular number of iterations (5 iterations in this work). The final shapes of the snakes are the shape estimate for the associated ‘cell candidates’. As mentioned previously, shape estimates are merged when they are overlapped and belong to a single object based on the merge scheme, and the merged shape estimate is the final shape estimate of the object in the ROI. In this work, two criteria are defined for the merge scheme based on the sizes and the orientations of the ACs.

The first merge criterion is that two snakes are merged when a large portion of one snake overlaps the other snake. The second criterion merges snakes when their orientations are close even if the area of overlap is small. Note that the orientation of a snake is represented by the direction of the major axis $\vec{d}_{E1}$. Suppose two snakes S_a and S_b have sizes $\{A_a, A_b\}$ and major axes $\{\vec{d}_a, \vec{d}_b\}$ respectively, and have an overlap area with size A_{ab} , the merge scheme is defined as in *Algorithm 2*, where $T_{A_{max}}$ and $T_{A_{min}}$ are the thresholds for the overlap ratio, and T_θ is the threshold for the orientation correlation. $T_{A_{max}}$ is the threshold for merging the snakes only based on sizes, and $T_{A_{min}}$ is the minimum overlap ratio that requires further consideration based on the orientation correlation in order to make the decision. In this work, $\{T_{A_{max}}, T_{A_{min}}, T_\theta\}$ are respectively assigned with the values 0.6, 0.1 and 0.7 (in radians).

Algorithm 2 Merge Scheme for Cellsnake

```

if  $\frac{A_{ab}}{A_a} > T_{A_{max}}$  or  $\frac{A_{ab}}{A_b} > T_{A_{max}}$  then
    merge  $S_a$  and  $S_b$ 
else
    if  $\left(\frac{A_{ab}}{A_a} > T_{A_{min}} \text{ or } \frac{A_{ab}}{A_b} > T_{A_{min}}\right)$  and  $\left|\left(\frac{\vec{d}_a}{|\vec{d}_a|}\right) \cdot \left(\frac{\vec{d}_b}{|\vec{d}_b|}\right)\right| > T_\theta$  then
        merge  $S_a$  and  $S_b$ 
    else
         $S_a$  and  $S_b$  belong to different objects
    end if
end if

```

6.5 Segmentation Results

The new AC model is firstly tested on some synthetic data with overlapped objects for ground-truth evaluation. An example of the synthetic data is given in Fig. 6.17(a) where each fibre is highlighted with different a colour. These fibres are extracted from the real data, and are classified as individual fibres in the images by the biologists (as pointed out by the arrows in Fig. 6.23). Each of the synthetic data is generated by manually overlapping the extracted single fibres.

The fitting results of the AC model and the segmentation results after merging are shown in Fig. 6.17(c) and (d). The new AC model fits each ‘cell candidate’ without deforming to the regions belonging to the other objects in the overlap regions, and the merge scheme merged the correct snakes for the corresponding fibres. The final shape estimates of Cellsnake approximate the shapes of the actual objects and separate them in the overlap ROIs.

Cellsnake is also tested with the real data and the shape estimates for the overlapped objects in the highlighted regions ‘A’, ‘B’ and ‘C’ of Fig. 6.1 are presented in Fig. 6.18 and Fig. 6.19. As shown in the figures, although the shape estimates of Cellsnake do not fit the exact shapes of the cell/fibre objects, they approximate the shapes of both the overlap and attached cell/fibre

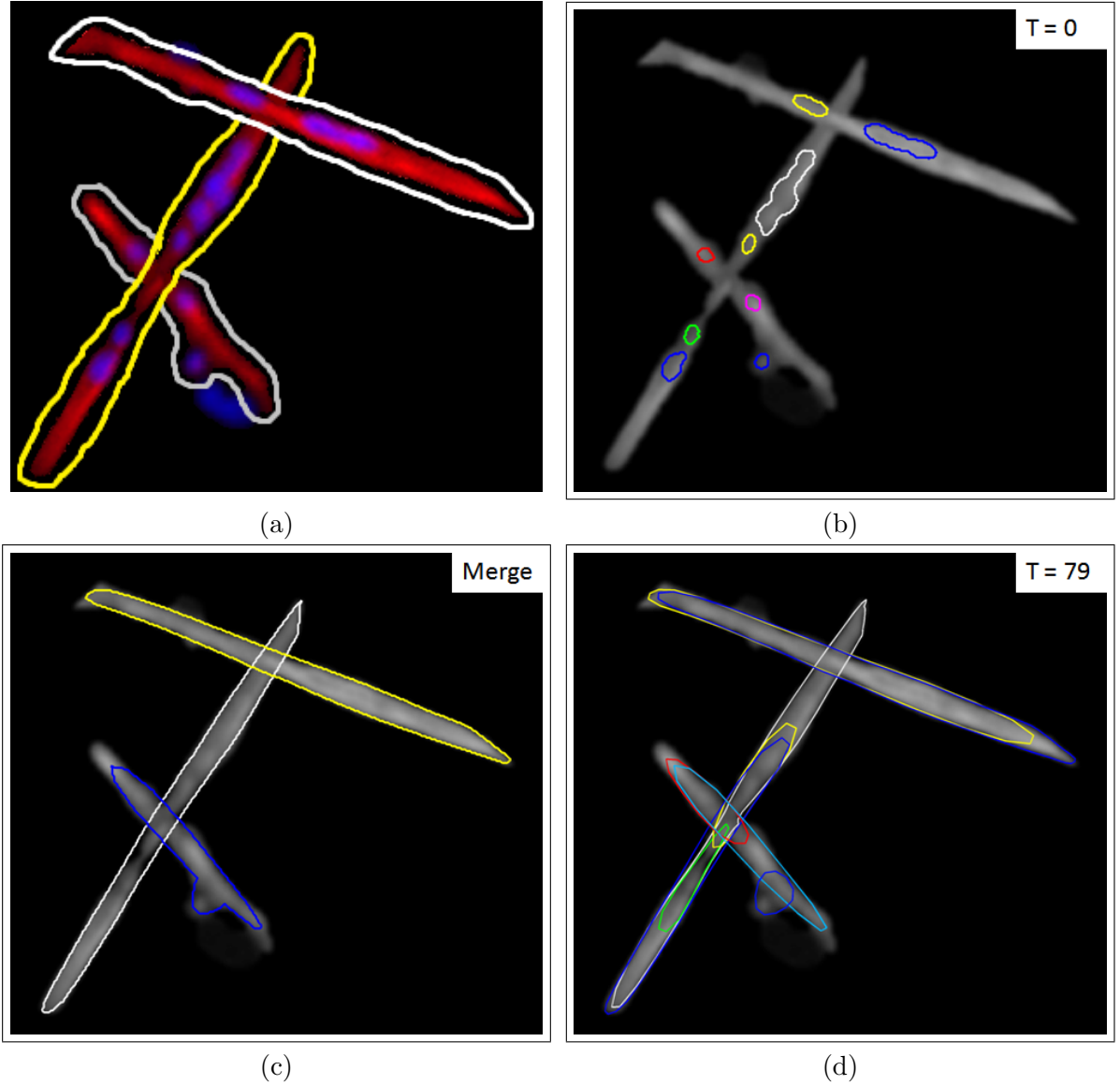


Figure 6.17: Example of synthetic overlapped objects and the shape estimate result (a) shows an example of synthetic overlap region generated by manually overlapping or attaching some non-overlapped cell/fibre objects ‘extracted’ from the real data, e.g. the individual fibres pointed out by the arrows in Fig. 6.23. (b) shows that the snakes are initialized as the boundaries of the nuclei regions. (d) shows the fitting results of the snakes. (c) shows the final shape estimates for the overlapped/attached objects after merging the corresponding snakes using the merge scheme.

objects which cover the correct nuclei for the corresponding cells/fibres. The shape estimation of the occluded fibres in the overlap region is constrained by the cellular module $\mathbf{F}_{\text{cell}}$ of the new AC model and hence prevents unwanted deformation (e.g. propagate to the regions belonging to other cells/fibres) in the occluded region. In order to show this constraint in detail, the deformation of the snake is illustrated for a number of iterations from the initialization ($T = 0$) to the termination (Fig. 6.20, Fig. 6.21 and Fig. 6.22 for region A, B and C respectively). The movies of the complete evolution can be found at <http://www.mee.tcd.ie/~sigmedia/Research/ICIP11Cellsnae>.

The elliptical force $\mathbf{f}_e$ helps the snakes to overcome the influence from the other objects when propagating through the overlap regions and prohibits the diffusion at those regions when the evolution continues. Also, the curvature force $\mathbf{f}_\kappa$ facilitates the deformation of the snakes through the narrow regions and allows long narrow fibres to be identified. Conversely, the centre shift force $\mathbf{f}_{cs}$ prevents the deformation of the small cells/fibres over to the adjoining adjacent objects. These snakes (the small single cells in region ‘A’ and the thin and elongated fibre in ‘C’) remain in the ‘attached’ objects and halt the propagation when they reach the ‘weak’ internal boundaries even though their growing forces are biased toward propagation of the curve beyond the boundaries. On the other hand, those big snakes that grow along the ‘weak’ internal boundaries are constrained by the smoothness forces and hence prevent internal leakage. Furthermore, although the snakes grow fast (large step size), the evolution of the snakes are ‘smooth’ (no sudden changes caused by the environment forces $\mathbf{F}_{env}$) during the deformation. This is due to the constraint of the momentum force $\mathbf{f}_m$ to ‘smoothen’ the evolution which also damps the ‘oscillations’ of the snake around the ROI boundary. Finally, the merge scheme groups the correct snakes (as shown in the last figure in Fig. 6.20 before merging and the merge results in Fig. 6.18) and hence provides sufficient shape estimates for the overlapped objects to cover the corresponding nuclei.

The combined segmentation results for the entire example image in Fig. 6.1 are shown in Fig. 6.23. The parameter setting in the model for most of the objects in the image are $\{w_e = 6, w_\kappa = 4, w_m = 2, w_{cs} = 3, k_{in} = 6, k_{out} = 8, \gamma = 1, \lambda = 0.5, a_M = 1000, \lambda_p = 0.1, \lambda_r = 0.1, a_\tau = 1\}$. For the objects with large overlap regions, w_e is essential and is set higher in order to enforce the elliptical shape and prevent the undesired divergence of the deformation in the overlap regions. For the very thin objects, w_κ is effective to help the snake to occupy the narrow regions. For typical ROIs that have adjoining objects, the parameter of the growth-control factor is increased to $a_\tau = 2$ so that the centre shift force $\mathbf{f}_{cs}$ becomes the dominant constraint force. Also, when a_τ is larger, the small snakes would be more sensitive to the ‘topology’ of the ROIs, and are easier to stop by the weak internal edges. Consequently, the small adjoining snakes are prevented from propagating to the bigger objects which can cause false merging in the merge stage.

Another sample image and the segmentation results are presented in Fig. 6.24. For this image, the model parameters are set as $\{w_e = 6, w_\kappa = 6, w_m = 2, w_{cs} = 3, k_{in} = 8, k_{out} = 10, \gamma = 1, \lambda = 0.5, a_M = 1000, \lambda_p = 0.1, \lambda_r = 0.1, a_\tau = 1\}$. Note that the weights for the curvature

force $\mathbf{f}_\kappa$ and the VFC force $\mathbf{f}_{\text{vfc}}$ are higher than the previous example. This is because the objects in this image are thinner and some of them are more ‘twisted’. For such objects, $\mathbf{f}_\kappa$ needs to be enhanced in order to enforce the snakes to propagate in the narrow regions which are restricted by the smoothness effect. Also, the amplification of $\mathbf{f}_{\text{vfc}}$ is to attract the snake (which is constrained by $\mathbf{f}_e$) to deform to the ‘winding’ regions such as the objects in at the bottom of the image. The reason for not reducing $\mathbf{f}_e$ is that the objects are thin and long, and the overlapped objects (such as the fibres at the centre of the image) require the elliptical effect for the snake to propagate over the overlap regions.

As shown in the results, Cellsnake can approximate the shapes of both the non-overlapped and overlapped objects which indicated by the ground-truth segmentation from biologists (Fig. 6.23 and Fig. 6.24). The shape estimations are reasonable as they retrieve the major shape appearance of the objects in the images and more importantly covers the correct nuclear regions for the corresponding cells/fibres which helps the biologists to access accurate cell/fibre density measurement from the skeletal muscle images as shown in Fig. 6.23 and Fig. 6.24. However, there is false merging for the two adjoining fibres at the top right of the image. This is because the orientations of two fibres are very close to each other. On the other hand, some fibres are not fully covered by the shape estimations when each of them has only one nuclei region but the cell-body (the ROI) has multiple branches. This is because Cellsnake is constrained by the elliptical force $\mathbf{f}_e$, and only one branch is chosen by the snake during the deformation in such regions. Furthermore, some single cell objects are falsely classified as fibres. This happens when the nuclear region is large. But the problem is not related to Cellsnake but instead is caused by the eSMEM shape modelling as the single nucleus is fitted by two or more components in the associated Gaussian mixture model.

The goal of this part of the work is to distinguish cell/fibre objects in a given skeletal muscle image. Cellsnake approximates the shapes of overlapped objects by merging the deformed ACs that belong to the same object, so that the shape estimation covers the correct nuclear regions for the corresponding candidates in the ROI, and hence the segmented candidates can be classified as single cells or fibres by the number of nuclei inside the shape estimation. Based on the cell/fibre classification, the biologists can then assess the cell/fibre density of the imaged testing sample. However, the models relies on a large number of parameters. With fixed parameters, Cellsnake can provide sufficient shape estimations for the simple cell objects and most of the fibre objects in a given image that satisfy the biologists to assess the cell/fibre density, but the objects with relatively large overlap regions are the major concerns for the model and usually require adjustment of the model parameters for better fitting. This also happens for the objects with very complicated shapes. A possible improvement for the model in such cases is to allow the model to alter the parameters automatically based on the shape of the snake. Also, there are insufficient images available in this part to evaluate fully the performance Cellsnake, and hence further investigation is required to examine and improve the robustness of the model for different complex shapes. Furthermore, a few false fibre estimations are caused by the merging

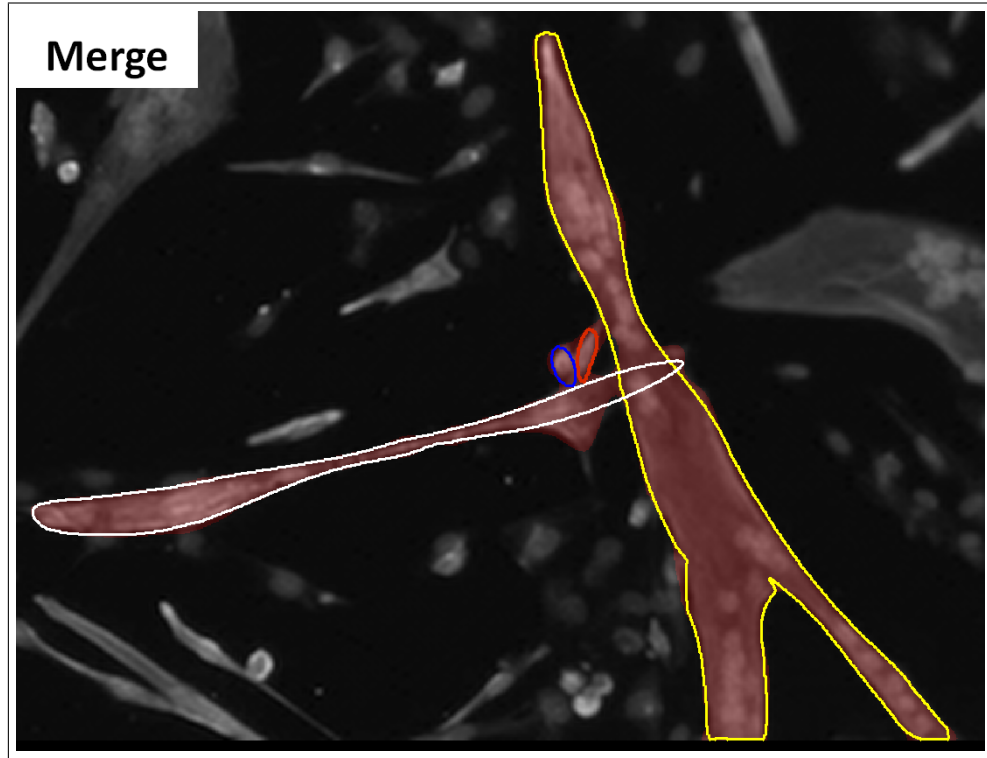


Figure 6.18: The final segmentation results after merging the ACs following the merge scheme. (For region A in Fig. 6.1)

criterion rather than the AC model, this may be improved by employing a more flexible criterion that considers the image features of the objects, and the false classification of the single cells as fibres which is caused by the estimation of Spotnick may be solved by constraining the GMM shape modelling of the nuclei with the shape estimate of cells.

6.6 Conclusion

This chapter outlines a new active contour mode (Cellsnake) for shape estimation of the overlapped cell/fibre objects and hence detect the objects from the combined appearance in the image. The goal of this part is to count cell and fibre objects in an image in order to help the biologists to assess the cell/fibre density from the image. Spotnick can be used to count Nuclei and if an object has more than one nucleus it is a fibre. However it is necessary to determine which object each nucleus belongs to. This is difficult because objects overlap each other making them difficult to delineate. Cellsnake attempts to achieve this using active contours to delineate the object boundary and uses novel forces and a novel merge scheme to overcome the overlap problem. Hence, the solution to this task is to combine Spotnick and Cellsnake as a joint decision.

The difficulty in the cell body segmentation is that the combined objects (with overlapped

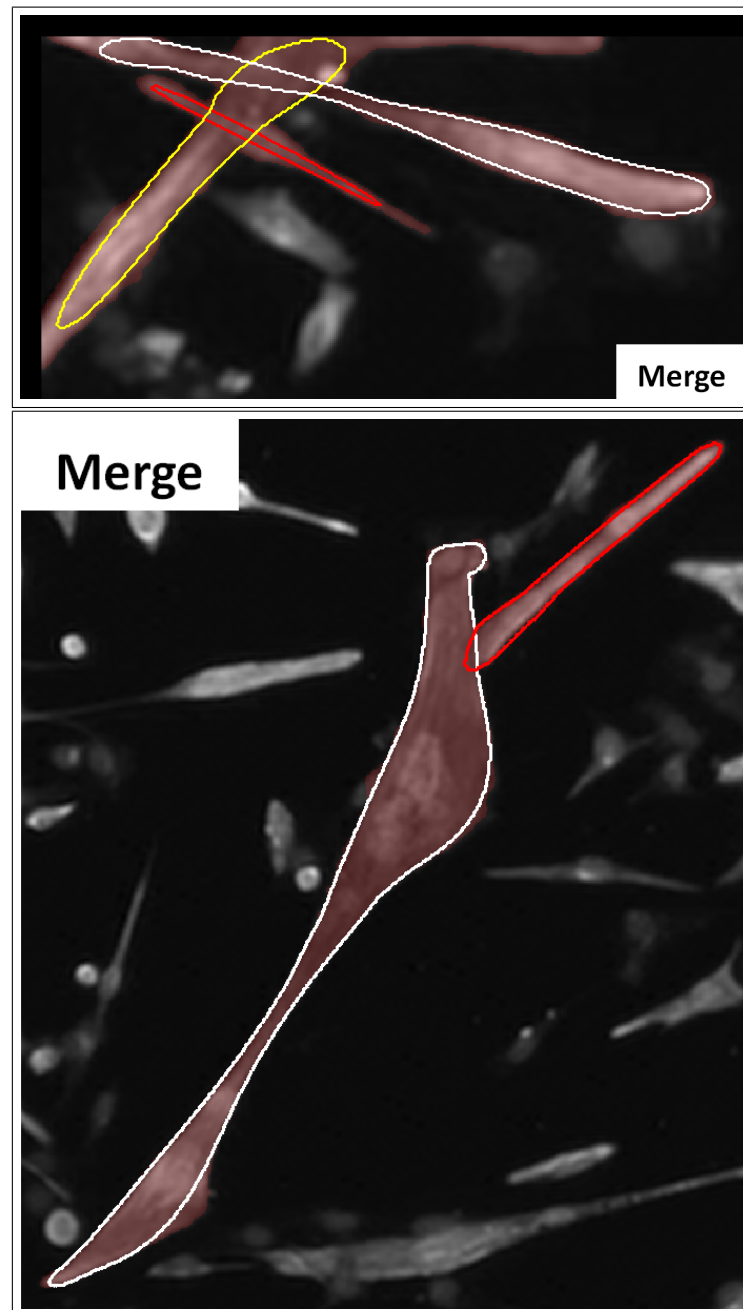


Figure 6.19: The final segmentation results after merging the ACs following the merge scheme.
(For region B and C in Fig. 6.1)

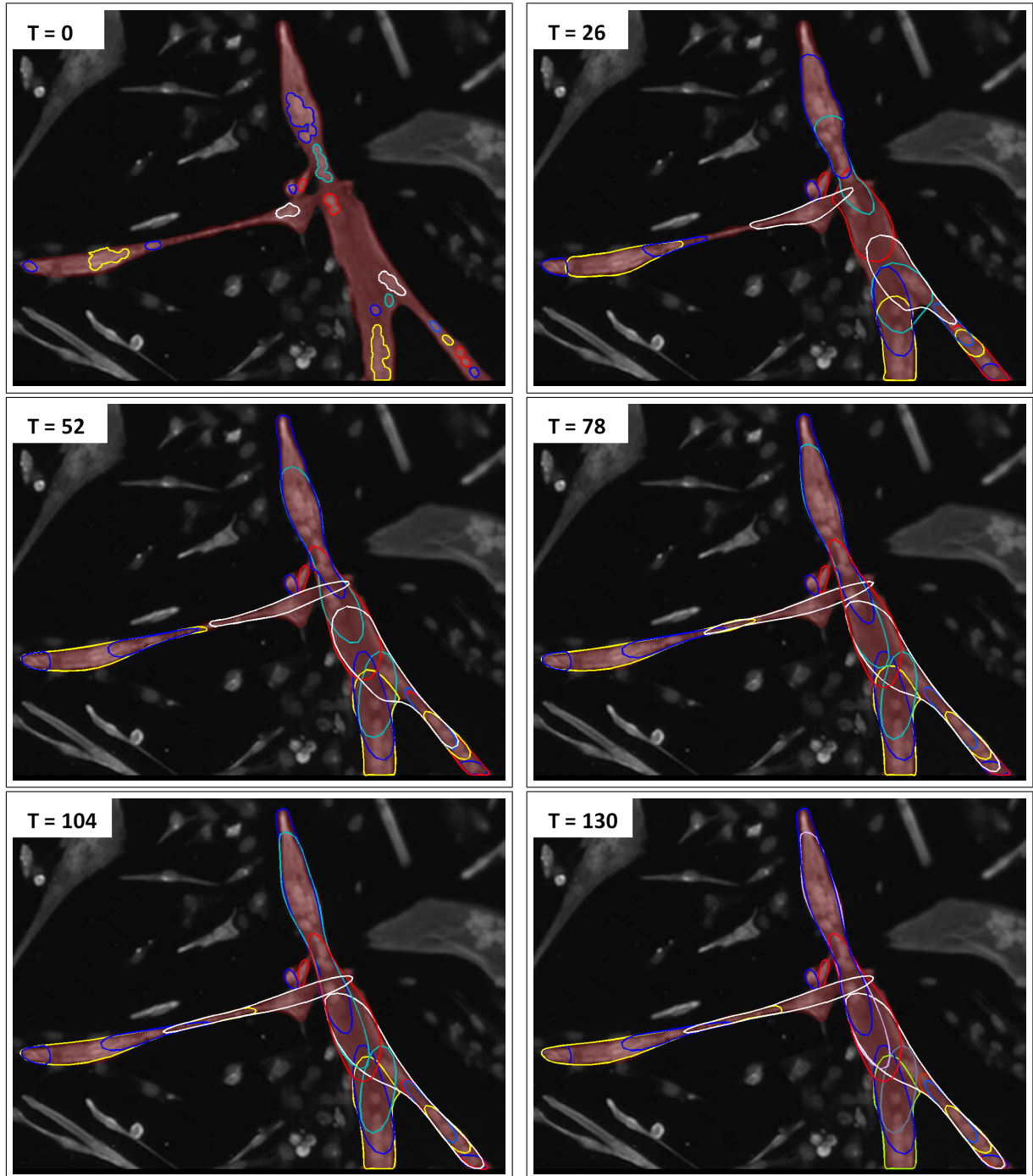


Figure 6.20: Illustration of snake deformation in region A of Fig. 6.1.

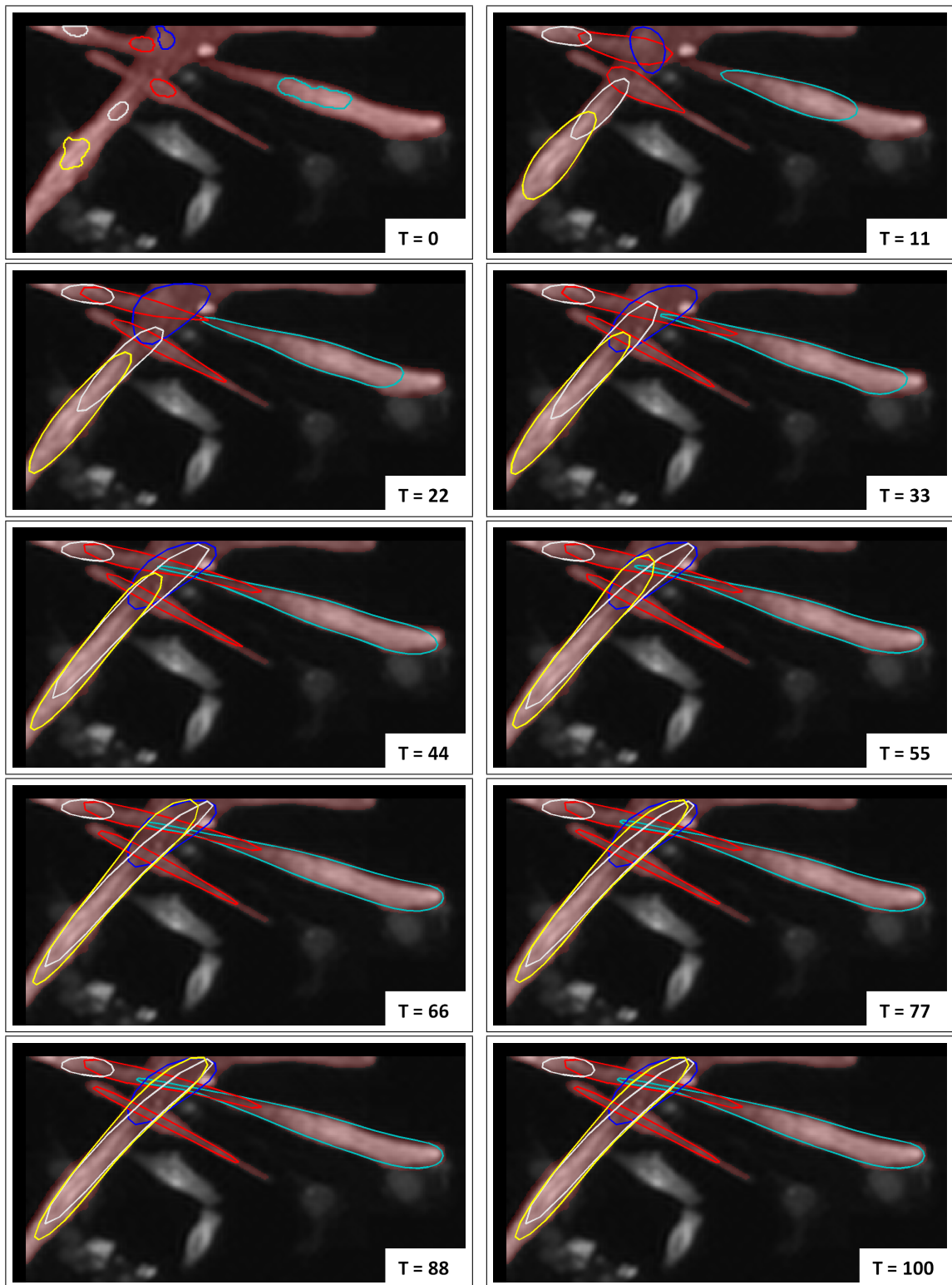


Figure 6.21: Illustration of snake deformation in region B of Fig. 6.1.

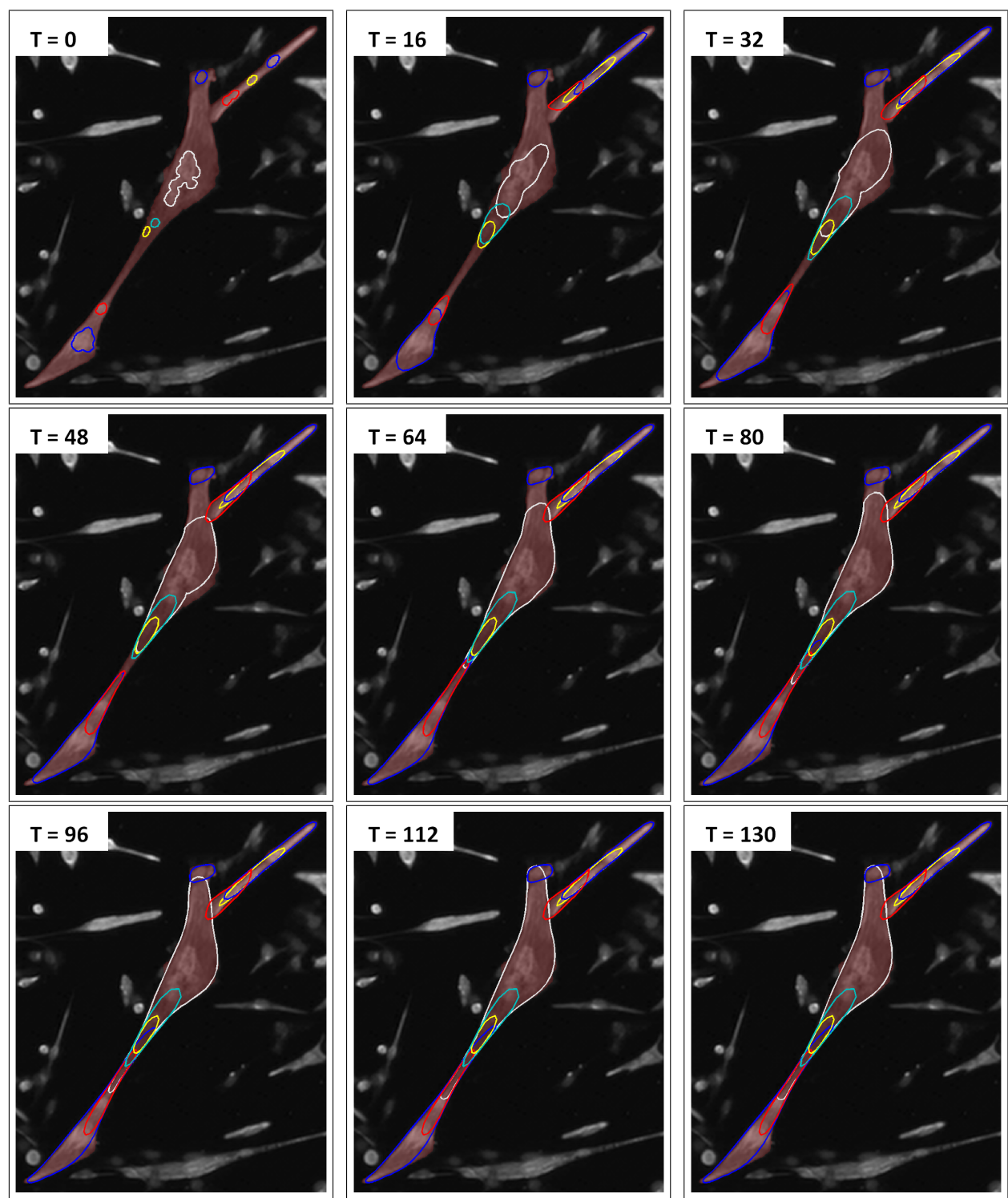


Figure 6.22: Illustration of snake deformation in region C of Fig. 6.1.

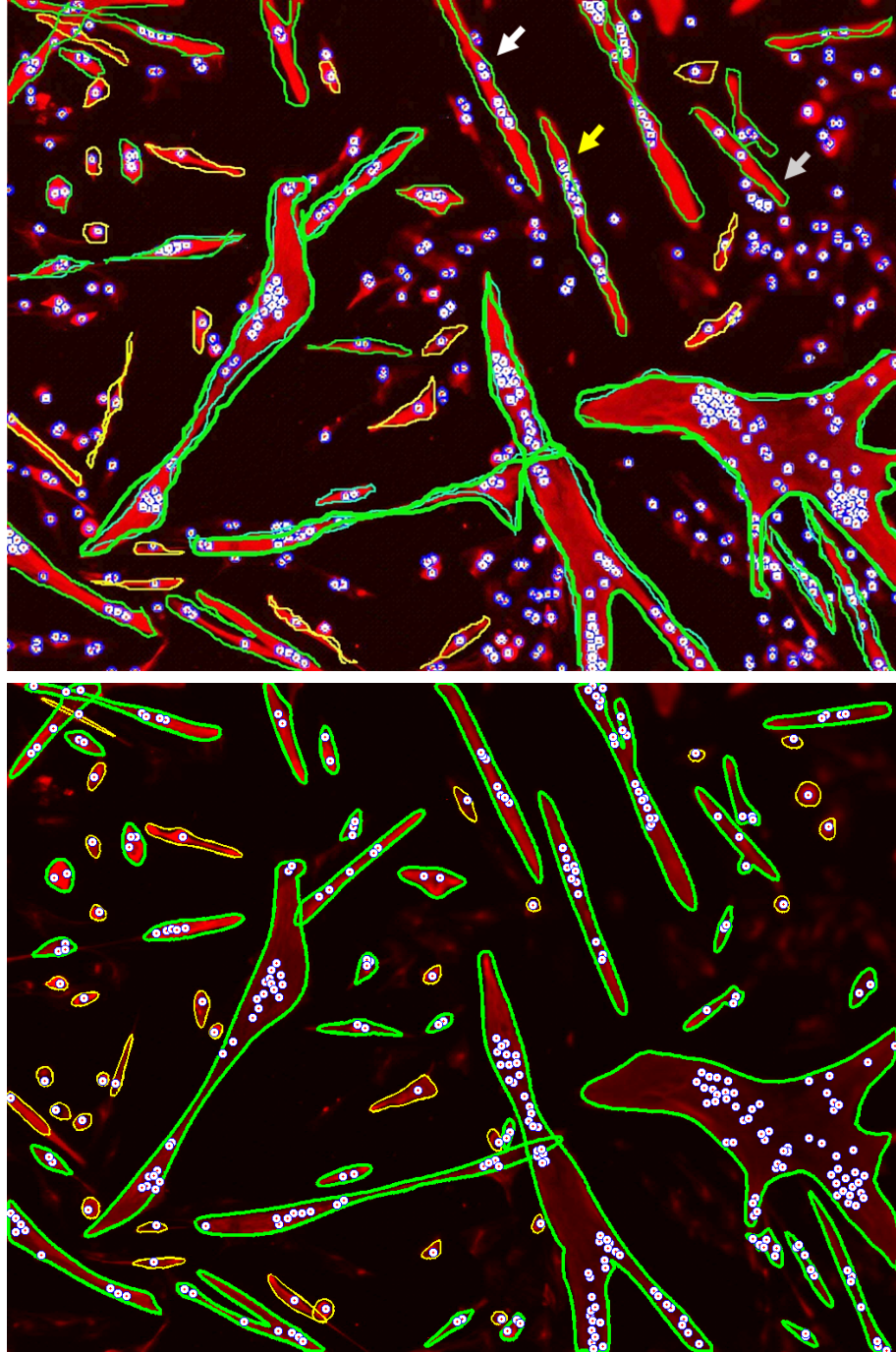


Figure 6.23: The complete segmentation results for the given image in Fig. 6.1 (The top image shows the ground-truth segmentation and classification of the cells/fibres from the biologists. The bottom image shows the segmentation results from Cellsnake. The white objects with blue circles are the separated nuclei estimated by the eSMEM shape modelling described in Chapter 3. The objects highlighted by yellow line and green line are the segmented single cells and fibres respectively. The arrows point out the individual fibres that are used to generate the synthesized data shown in Fig. 6.17)

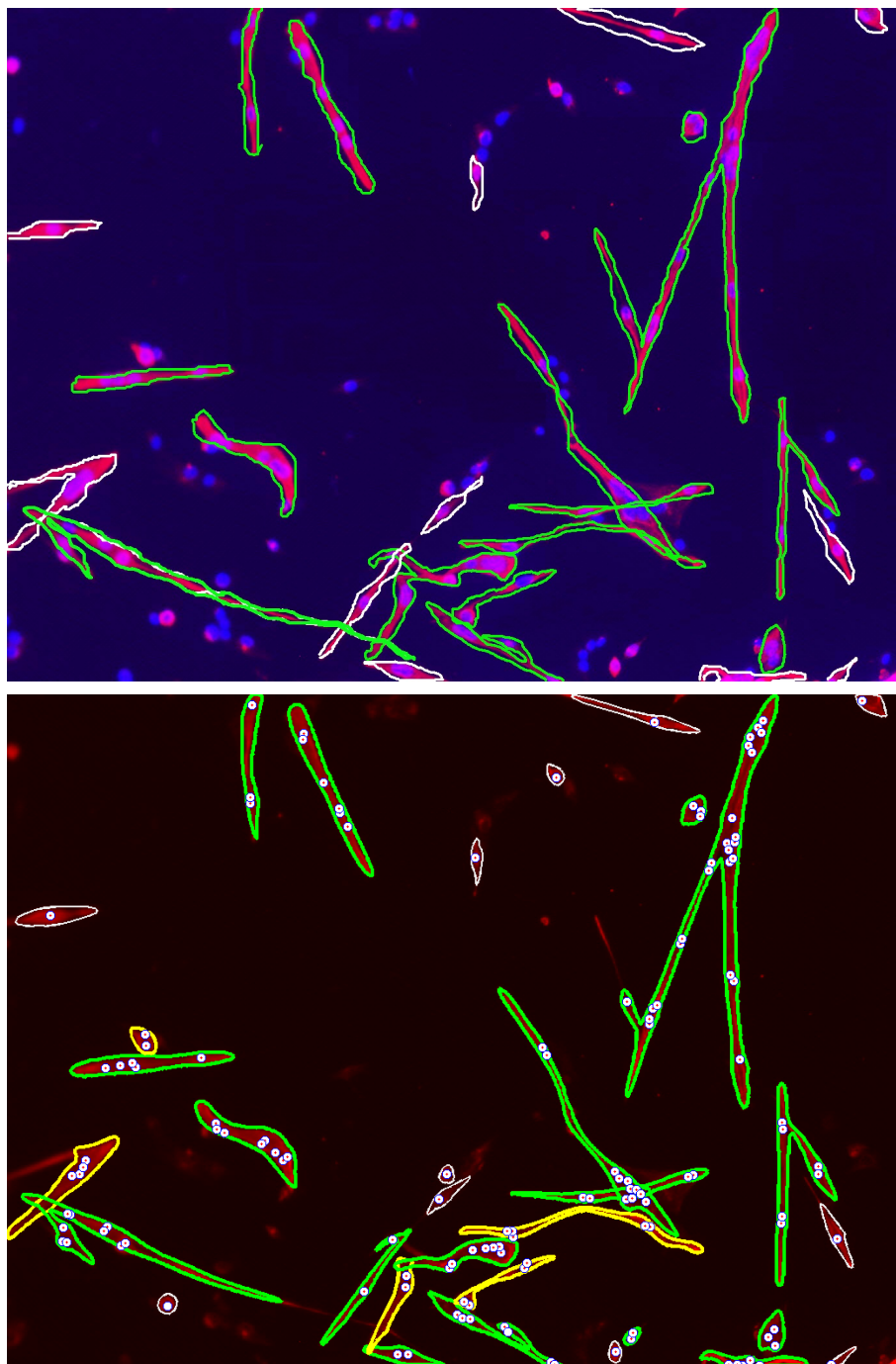


Figure 6.24: Another example of the complete segmentation (*The top image shows the ground-truth segmentation and classification of the cells/fibres from the biologists. The objects highlighted by white line and green line are the segmented single cells and fibres respectively. The bottom image shows the segmentation results from Cellsnake. The fibre estimates are highlighted by yellow lines and green lines. The white objects with blue circles are the separated nuclei estimated by the eSMEM shape modelling described in Chapter 3.*)

cells/fibres) and the fibres both have irregular shapes, the detection of the overlapped objects in the ROIs becomes complicated when it is difficult to distinguish the combined objects and the fibres. The decision of whether a ROI is a single fibre or a group of overlapped fibres/cells is based on how the non-overlapped parts in the ROI are ‘connected’ through the overlap regions. Also, the purpose of this task is to count the number of cells and fibres in the image, which are differentiated by the number of nuclei inside the cell-bodies, it is more important that the shape estimations of the overlapped candidates cover the correct nuclear regions for the corresponding cell/fibre, rather than getting the exact boundaries of the objects. Although the fibres have complicated shapes, each of them is formed by the merging of cells which have simpler and more regular shape patterns. Therefore, the idea of Cellsnake is to use multiple small ACs to fit different parts of the ROI and merge the ACs that belong to the same cell/fibre object after the deformation. Since the key of the classification between cells and fibres is the number of nuclei they have, each of the small AC is assigned to each separate nucleus cluster in a given ROI. The small snakes are considered as ‘cell candidates’ in the ROI, and the deformation of these hypothetical cells is regulated by a cellular shape functional so that the snakes are constrained by the non-overlapped parts of the ROIs when the snakes propagate through the overlap area. The designed merge criteria ‘rejoin’ the connected snakes based on the orientations of the ACs and the overlap area compared to the inner area of the ACs, so that connected ACs approximate the shapes of the corresponding real cell/fibre candidates in the ROI, and at the same time cover the appropriate nuclei regions where they are initialized.

The complete technique is tested with synthetic combined objects containing manually overlapped single objects, and is also tested with real data. However, the weakness of Cellsnake is that the different cellular constraints in the model are balanced by a number of regularizing parameters. With fixed parameters, Cellsnake can provide sufficient shape estimations for the simple cell objects and most of the fibre objects in a given image that satisfies the biologists requirement to assess the cell/fibre density, but the objects with relatively large overlap regions are the major concerns for the model and usually require adjustment of the model parameters for better fitting. This also happens for the objects with very complex shapes. One of the issues with the evaluation of Cellsnake is the lack of test images available and further investigation is required to examine and improve robustness of the model for different complex shapes. Also, a few false fibre estimations are caused by the merging criterion rather than the AC model, this may be improved by employing a more flexible criterion that considers the image features of the objects. Furthermore, since the number of nuclei in each nuclear region is estimated by using Spotnick described in *Chapter 4*, the false classification of the single cells as fibres can possibly be improved by constraining the GMM shape modelling of the nuclei with the shape estimate of cells.

7

Conclusion

This thesis has investigated the difficulties of microscopic image segmentation in the neuro-study of memory formation [46] and investigation of a novel muscle regeneration technique [40]. The demands of image processing from the biological researches includes: 1) the extraction and colocalization analysis of protein particles; 2) 3-D shape reconstruction of neuron synapses; 3) segmentation of overlapped muscle fibres/cells. Since the structures of the target objects and the conditions in the three applications are very different to each other, this thesis has introduced three novel shape-based segmentation algorithms to tackle the different problems independently. The following sections present a brief overview of the work in this thesis and propose some possible improvements and ideas for the future work.

7.1 Microscopic Image Segmentation

A brief overview of the state of the art in microscopic image segmentation is firstly presented in *Chapter 2*. The different previous techniques proposed for a wide range of microscopic images are categorized as image driven approaches and model driven approaches. The image driven approaches operate directly on the image pixels and detect objects solely based on the image features, while the model based approaches assume that objects in an image are presented in a certain pattern. Since the usage of prior shape information provides high accuracy and possible shape estimation for the target objects, model based approaches have drawn a large attention and have increasing preference in microscopic image segmentation. The algorithms introduced in this thesis are therefore model based approaches. The models developed based on

the shape information of the different types of structures. Also, as the image data for the work in this thesis are collected from confocal microscopes, a brief description of the mechanisms of fluorescent confocal microscopy is presented in this chapter.

7.2 Spot Detection and Analysis

A novel Gaussian mixture shape modelling algorithm is developed in *Chapter 3* for extracting the protein particles in the images delivered from the biological study of memory formation. The study is to investigate the roles of the different types of proteins in the protein synthesis governing the formation of long-term-memory. A brief biological background is presented, and the image processing task is to extract the protein particles and analyse the colocalization between the different types of detected protein particles. In order to detect the candidate particles in each cluster-spot (cluster of particles) that has a merged appearance in the images, the introduced shape modelling algorithm fits the intensity profile of each spot of the protein particles (some are cluster-spots) with a 2-D Gaussian mixture model (GMM). The new GMM approach not only estimates the number of candidate particles in a cluster spot, but also parameterizes the shape characteristics of the particles for the later co-localization analysis.

Since the optimization process for the parameters of the GMMs is intractable, a novel error-based split-and-merge expectation-maximization (eSMEM) algorithm (as part of the shape modelling algorithm) is applied to estimate the parameters of the mixture components of the model as well as the number of components. Unlike the conventional split-and-merge expectation-maximization (SMEM) algorithms for data point clustering (which are briefly reviewed in this chapter), eSMEM is specialized for shape modelling and is deterministic, i.e. repeatable.

In order to overcome the scanning-noise and the background luminance problems in the real image data that can reduce the modelling accuracy, a 3-step pre-processing stage of the shape modelling algorithm is proposed in *Chapter 4*, and the complete shape modelling algorithm is named ‘Spotnick’. The shape modelling compatibility and the performance of the split-and-merge operation of Spotnick are firstly examined by various synthetic testing models, and Spotnick gave good modelling results. Also, the eSMEM algorithm is compared with the Zhang *et al.* SMEM algorithm on synthetic Gaussian mixture models with non-overlapped or overlapped components in different model orders. Since the real data has background noise, the two algorithms are also tested by the synthetic models with Gaussian background noise in the images of the models. The comparison shows that the eSMEM algorithm has better performance in shape fitting, model order detection, position estimation and size/shape estimation, and improvement in performance is greater for larger number of mixture components. The results also show that the importance of the prioritized split-merge selection stage for the eSMEM algorithm to prevent over-fitting for the spot analysis, and also shows that the proposed initialization method helps the algorithm to achieve better modelling results.

On real data, Spotnick provides approximations to the ‘merged’ spots and is sufficient to

parameterize the candidate particles in the cluster spots for the co-localization analysis. After showing the shape modelling results, the usefulness of the results to the biological analysis is demonstrated by showing how the shape models are used to perform data analysis and co-localization analysis on the protein particles in the images. In reality, the tools of Spotnick (as described in *Appendix B*) are being utilized in the neuro-study of memory formation to analyze the co-localization of the different proteins and has already made contribution to the biological study. The biologists have found early observations and characterization of candidate synaptic mRNP particles that contain Me31B, and have suggested a simple model in which Me31B mediates the formation of synaptic mRNP particles that contain locally repressed mRNAs in [46].

7.3 3-D Shape Reconstruction

In *Chapter 5*, a Bayesian segmentation model is introduced for the 3D reconstruction of neuron synapses. Since the protein synthesis for memory formation takes place at the synapses of the neuron cells, the shape information of the synapses for where the protein particles are located during protein synthesis is important for the biological investigation. As the synapses are distributed in 3D in the specimen, the shape information of the synapses is captured by scanning the synaptic object at multiple different ‘depths’ from top to bottom of the living specimen, and is stored in the image stack of the serially scanned slices.

The task is to reconstruct the 3-D shape of the synapses from the observed image stack when a degree of out-of-focus detail is still present in each image. Due to the out-of-focus problem, the boundaries of the synaptic structures are difficult to define in the images. The goal is to distinguish the in-focus synaptic features from the out-of-focus features. However, the segmentation becomes complicated when the luminance of the synaptic object is irregular which is caused by the defect of the fluorescent staining, where the fluorescent markers are distributed unevenly over the synapses. Since the decision of detecting the in-focus features in this case is based on the sharpness of the features, the luminance of each feature across the slices, and the priori connectivity information of the synaptic structure, a Bayesian inference model named Synapcut is proposed in this chapter to integrate the different information and segment the object based on decision making.

There are three likelihoods and one frame prior in this model. The ‘sharpness’ information of the synaptic features is encoded by the high frequency content of the image in the wavelet domain. This gives rise to an associated wavelet feature for each pixel which is a vector of wavelet coefficients in the high frequency sub-bands of the image, and the first likelihood (the wavelet likelihood) for each pixel is estimated by the associated wavelet feature. Since the wavelet likelihood is insensitive to the luminance, the second likelihood (the global intensity likelihood) is applied to eliminate background noise that may have similar wavelet features to the synaptic objects and cause false alarms. The brightness information of the in-focus features is derived from the inter-slice intensity as the intensity of each in-focus object decrease when it

becomes out-of-focus. Hence, the third likelihood (the inter-slice intensity likelihood) is applied to select the in-focus features from the potential synaptic features by considering the inter-slice intensity profile of each synaptic branch across the slices. The likelihoods are incorporated with a frame prior that enhances the connectivity of the synaptic features. The segmentation model is optimized using the graph-cut method.

Synapcut is successful in segmenting the segments of obvious in-focus synaptic features and also detects connectivity of the features. However, it is difficult to evaluate the complete segmentation precisely, as the ground-truth segmentation of real data is generated manually by the biologists. The manual evaluation of the whole 3D segmentation is difficult to assess by the biologists. This is because the manual segmentation of the synaptic features is only based on the image data of the current slice. The 3D segmentation requires the consideration of the feature data across the slices, especially when the orientations of the features do not align to the plane of the slices, but this is difficult and very time consuming to achieve. Since the parameter adjustment for Synapcut is based on manual evaluation, it is difficult to determine optimal values of the model parameters. Hence, further investigation is required for validating the model and finding the optimal model parameters.

7.4 Segmentation of Overlapped Objects

Chapter 6 introduced a new active contour mode (Cellsnake) for shape estimation of the overlapped cell/fibre objects in order to segment the objects from the combined appearance in the image. The goal of this work is to detect the cell and fibre objects from the given skeletal muscle images where some of the cell and fibres are overlapped and appear as combined objects (the regions of interest, ROI). The challenge of the detection arises from the combined objects which can be confused with the fibres with various shapes and each of the objects is falsely detected as a fibre. The decision of whether a ROI is a single fibre or a group of overlapped fibres/cells is based on how the non-overlapped parts in the ROI are ‘connected’ through the overlap regions. Also, the purpose of this task is to count the number of cells and fibres in the image, which are differentiated by the number of nuclei inside the cell bodies. Therefore, it is more important that the shape estimations of the overlapped candidates cover the correct nuclear regions for the corresponding cell/fibre, rather than getting the exact boundaries of the objects. The solution to this task is to combine Spotnick and Cellsnake as a joint decision, where Spotnick is used to count nuclei with GMMs and Cellsnake delineate the object boundaries for both overlapped and non-overlapped cell body objects using novel forces and a novel merge scheme to overcome the overlap problem.

Motivated by the fact that the fibres are formed by the cells and the cells have simple and regular shape patterns, Cellsnake fits a snake to each ‘cell candidate’ in the ROI and regulates the snakes to deform like hypothetical cells. These ‘cell candidates’ are represented by the separated nuclear regions (clusters of nuclei) in the ROI. The deformation regulation is derived from the

cellular patterns in the image and is defined in terms of four cellular constraint forces proposed in the AC model. Because the single cells tend to be thin and elongated, the major constraint force is the elliptical force that facilitates the deformation at one direction and restricts along the perpendicular direction. The other forces are derived by the curvature and the deformation momentum of the snake as well as the constraint of enforcing the centre of the snake at the centre of the corresponding nuclear region. The purpose of these forces is to use the non-overlapped parts of the ROI to regulate the propagation of the snakes over the overlap region to connect other snakes. After the deformation, the designed merge criteria merge the connected snakes based on the orientations of the ACs and the overlap area compared to the inner area of the ACs, so that connected ACs approximate the shapes of the corresponding real cell/fibre candidates in the ROI, and at the same time cover the appropriate nuclei regions from which they are initialized.

The complete technique is tested with synthetic combined objects containing manually overlapped single objects, and is also tested with real data. The results show that Cellsnake is sufficient to approximate the shapes of both the non-overlapped and overlapped objects which are indicated by the ground-truth segmentation from biologists, as shape estimations retrieve the major shape appearances of the objects in the images and cover the correct nuclear regions for the corresponding cells/fibres. By joining the estimations of Cellsnake and Spotnick, each estimated cell body object is classified as cell or fibre based on the number of estimated nuclei inside the object. This helps the biologists to access the fibre/cell density based on the numbers of the estimated cells and fibres from the image and hence perform further biological analysis.

However, the model relies on a number of regularizing parameters to control the effects of different constraints in order to suit the fibre objects with different complicated shapes in different images. With fixed parameters, Cellsnake can provide sufficient shape estimations for the simple cell objects and most of the fibre objects in a given image, but the objects with relatively large overlap regions are the major concerns for the model and usually require adjustment of the model parameters for better fitting. This also happens for the objects with very complex shapes. Also, a problem for the evaluation of Cellsnake is the lack of test images available, and hence further investigation is required to examine the robustness of the model and improve the model for different complex shapes by allowing the model to alter the parameters automatically based on the shape of the snake. Finally, the false fibre estimations caused by the merging criterion rather than the AC model may be improved by employing a more flexible criterion that considers the image features of the objects, and the false classification of the single cells as fibres which is caused by the estimation of Spotnick may be solved by constraining the GMM shape modelling of the nuclei with the shape estimate of cells.

7.5 Future Work

Microscopic imaging and image processing are of increasing interest to the scientific and engineering communities. Recent developments in cellular-, molecular- and nanometre-level technologies have enabled important discoveries in biology, medicine, chemistry, pharmacology, and many related fields. Associated with this research, there has been an ever-increasing volume of microscopy image data, and there are driven demands for automatic or semi-automatic image processing algorithm. This section outlines some possible improvements to the algorithms presented in this thesis to increase the robustness and automation, and discusses the potential applications of these techniques.

Gaussian mixture model for shape modelling

The GMM shape modelling algorithm ('Spotnick') provides appropriate estimates to the 'merged' spots in the protein images. In order to further assist the biological studies, a user interactive GUI is required that allows the user to remove undesired particle estimates (ie. the components in GMM) from the cluster-spots, or to add more particles to the estimation and re-estimate the new GMM by constraining the number of components.

Also, the modelling estimate can be improved by constraining the model parameters during the optimization process based on detailed shape characteristics of the target objects. Besides to understand the physical properties of the objects, shape information could be acquired from multiple slices of objects captured at different depths of the specimen under observation. This also leads to a possible extension of the GMM for 3D reconstruction of clustered spots where the 3D shapes of target microscopical objects are fitted by 3D Gaussian mixture models using the 3D morphological information in multiple slice images of the objects.

Finally, the future work in the area of shape modelling is to increase the robustness of shape fitting algorithm for a wider range of spots that have different shape appearance. In such cases, the modelling algorithm requires modification to incorporate different mixture models for the spot objects with non-Gaussian shapes, and the eSMEM algorithm also needs to be altered to fit the non-Gaussian models.

Wavelet-based Bayesian model for 3-D segmentation

The segmentation results presented in this thesis demonstrate the potential of the wavelet-based Bayesian model (Synapcut) for 3D reconstruction of complicated objects. However, this technique is in its infancy as it is unclear whether the parameter values described in *Chapter 5* are optimal for the model, or the optimal values are different and specific for different image stacks. This is an area that requires further research, and ideally, a complete ground-truth segmentation of a whole image stack would be essential for finding the optimal parameter values and evaluating of the performance of Synapcut. As there is no certain threshold value or criterion

for distinguishing the in-focus and out-of-focus, the evaluation for the segmentation result is subjective. This may require a statistical evaluation from a number of researchers that have certain knowledge about synapses and confocal microscopy. However, manual classification of in-focus features requires the consideration of the different projections of the same image feature at different adjacent slices in the stack. Hence, a GUI that allows the researchers to visualize the both the sample and the segmentation side by side and in both 3D (the 3D structure) and 2D (cross-section of the 3D structure) would be useful. The future work also includes the possible improvement of the inter-slice likelihood model in Synapcut based on the investigation of actual point-spread-function of the confocal microscope which would possibly improve the detection of in-focus features.

Active contour model for overlapped object segmentation

The performance of Cellsnake for overlapped cell/fibre segmentation gave encouraging results. The major concern of the model is the setting of the algorithm parameters as the optimal setting are varied depending on the variation and the complexity of the object shapes. Hence, the model in its current form is not well suited for assisting the biological research as there is a large number of user defined parameters associated with the model. A possible solution is to introduce a parameter adjustment framework to Cellsnake that automatically changes the values of the parameters during the deformation if needed. This automatic adjustment could be performed based on the consideration of the shape of the snake and the status of the snaxels.

In the area of the active contour (AC) models, the idea of Cellsnake is an interesting approach that fits the shape of each occluded object by constraining the deformation with shape patterns of the objects, especially when the shapes have large variation. The key is the elliptical forces that enable the AC model to have direction-dependent deformation. This idea can be extended to the other shape constraint rather than elliptical, and hence may become useful for the other applications rather than cell/fibre segmentation.

The future work for Cellsnake is to prove the usefulness and performance of the model by testing with more skeletal muscle images, and possibly develop the parameter adjustment framework in order to increase the robustness of the model or to make the algorithm simpler, use fewer forces and parameters. Also, the comparison with other AC models would be an important aspect of evaluating the performance of Cellsnake.

Final Thoughts

The three novel shape based algorithms developed in this thesis are assisting, and are also being tested by, the researchers involved in the neurological study of memory formation and the medical investigation of muscle regeneration techniques. These algorithms help the researchers to understand more and to get deeper insight of the subjects being investigated. The new information and findings of the target objects are valuable information for improving the algorithms

so that they can become more effective and efficient tools for biological and medical research.

Proceedings of the National Academy of Sciences of the United States of America



Protein Synthesis and Regulation in Neuron Cells

Protein synthesis is the process of producing protein molecules by translating the messenger ribonucleic acid (mRNA) with ribosome, where the mRNA is a molecule of ribonucleic acid (RNA) that is transcribed from a DNA template in the nucleus of a cell. Hence, the protein synthesis is equivalent to the translation of mRNA, and the translation begins while the mRNA is attached to a ribosome.

For memory formation, the mRNA is only translated at stimulated synapses of the neuron cell to form memory. This means that the mRNA translation is regulated during the transportation from the nucleus to the synapses as well as when stored at the synapses before stimulation. The regulation of mRNA translation is controlled by repression with some specific proteins, also known as repressors, which are binded to the mRNAs to resist the translational activities [6,16].

The process of repression is as shown in Fig. A.1. The transcribed mRNAs are firstly exported from the nucleus of the neuron cell (Fig. A.1(b)), and then immediately bound by specific proteins, i.e. the repressors, to repress the translational activities. Each repressed mRNA is bound by the ribosomal subunits and some other protein particles to form a messenger ribonucleoprotein particles (mRNPs), and sequestered into a macromolecular structure and form the RNA granules (Fig. A.1(d)). The RNA granules are then transported to and stored in the synapses of neuron cell, and await stimulation at the synapses (Fig. A.1(e)). The translational repression remains on the RNA granules until the resident synapses are stimulated. When the synapses are stimulated, the repressed mRNAs are released from the repressors by phosphorylation and lead to the local translation preformed by the attached ribosomal subunits.

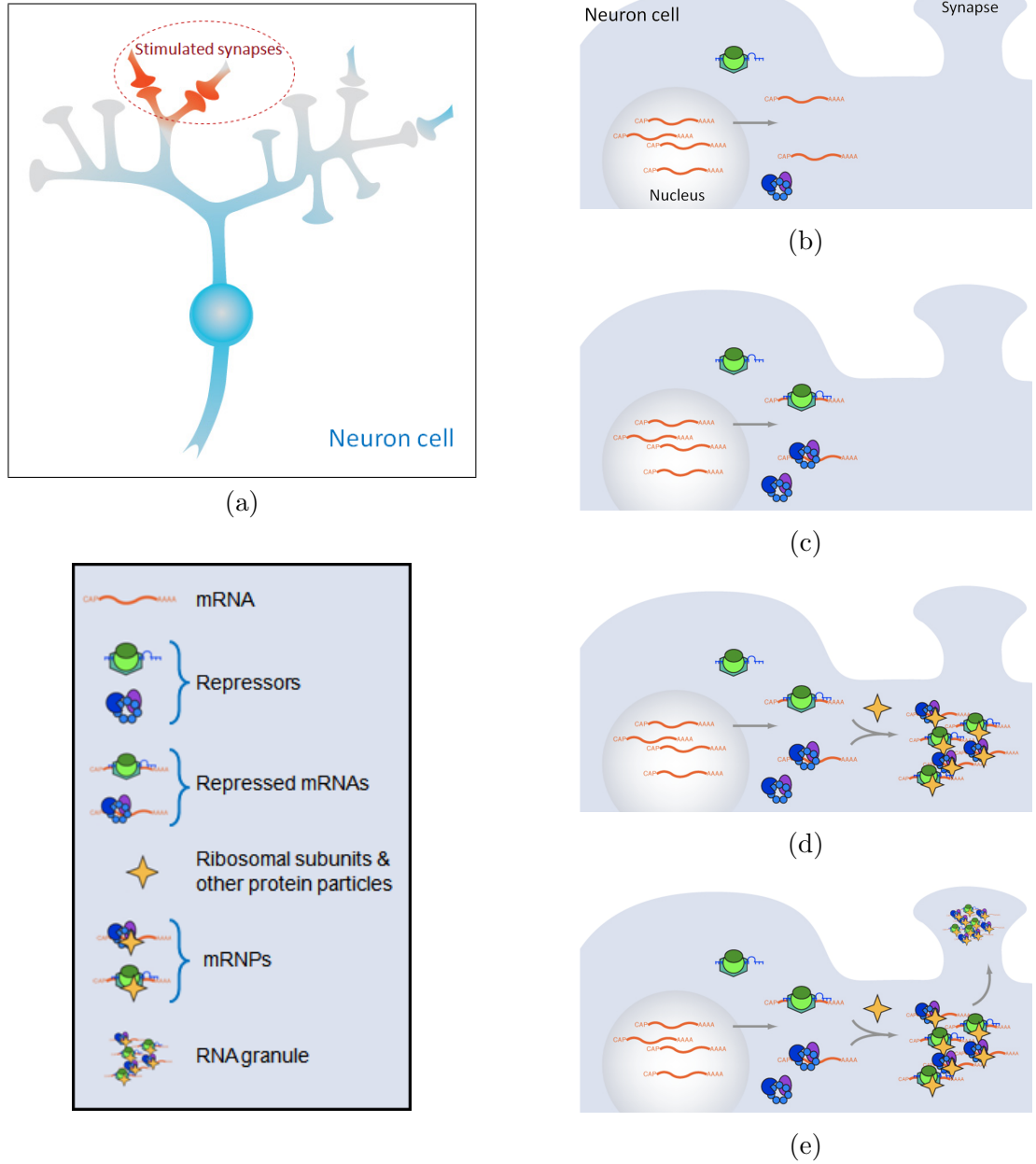


Figure A.1: Repression for mRNA translation in memory formation. (a) Memory formation is modulated by protein synthesis, i.e. mRNA translation, at the stimulated synapses (red) of neuron cell. The translation of mRNA is regulated by the repression which the procedures are illustrated in (b)-(e). (b) the transcribed mRNA is firstly emitted from the nucleus; (c) translation of mRNA is inhibited by binding with some specific repressing proteins, i.e. the repressor; (d) each repressed mRNAs is then bound by the ribosomal subunits and some other proteins to form mRNP particle, and sequestered into a RNA granule; (e) the RNA granules are transported to and stored at synapses.

B

Spotnick Graphical-User-Interface (GUI)

The presented Gaussian shape modelling algorithm is implemented as a GUI in MATLAB, in order to design to provide an intuitive interface for the user. In addition to the modelling, the GUI contains the following features: 1) image region selection; 2) contrast mask generation by the user defined thresholding; 3) estimation results and parameters statistics; 4) visualization of user filtered models; 5) co-localization analysis of the particle estimates after shape modelling.

Fig. B.1 shows the primary GUI of the program which allows the user to choose a test image and select region of interest in the image as an input of the program. Fig. B.2 shows the second GUI that allows the user to define the threshold for thresholding the modified contrast values of selected images and hence segment the spots of interest from the images. The mask, also the contrast mask, results from thresholding is drawn on top of the original image of the corresponding channel in the given image. The third GUI allows the histograms (Fig. B.3) of the data estimated from the shape modelling process, and the combined modelled image to be superimposed on top of the input image (Fig. B.4). The image (Fig. B.5) that contains the filtered modelled data can be previewed after setting the thresholds for the size, peak intensity and the background intensity. Also, the co-localization analysis (Fig. B.6) of the filtered modelled data is accompanied with the preview. The last GUI shown in Fig. B.7 is used for averaging the co-localization analysis of a series images, and analysis of each image can be reviewed by selecting the image title in the list block.

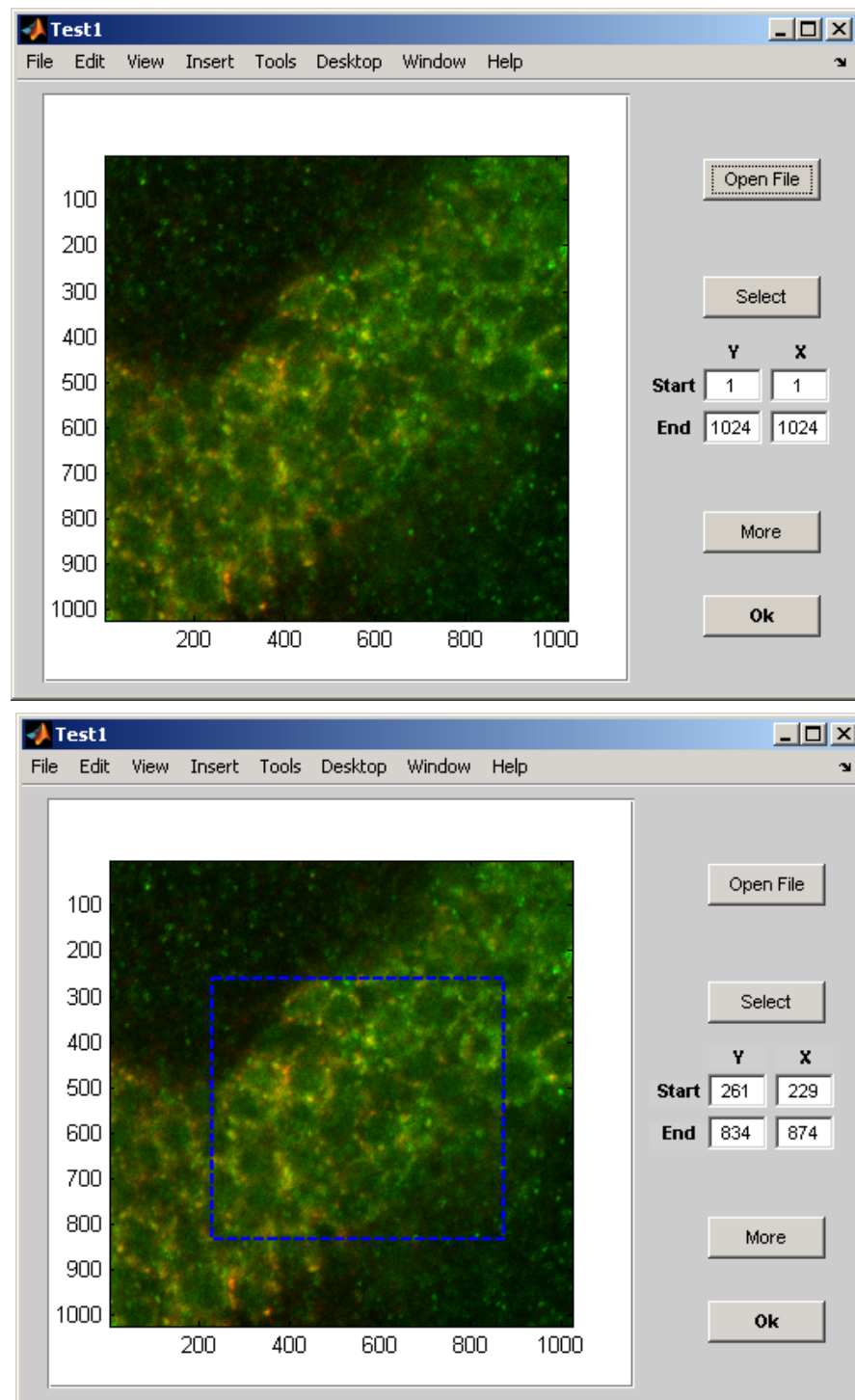


Figure B.1: Test1 (This GUI allows the user to select the image for analyzing. The user can also specify a (blue) rectangle ROI on the selected image for analyzing rather than the whole image.)

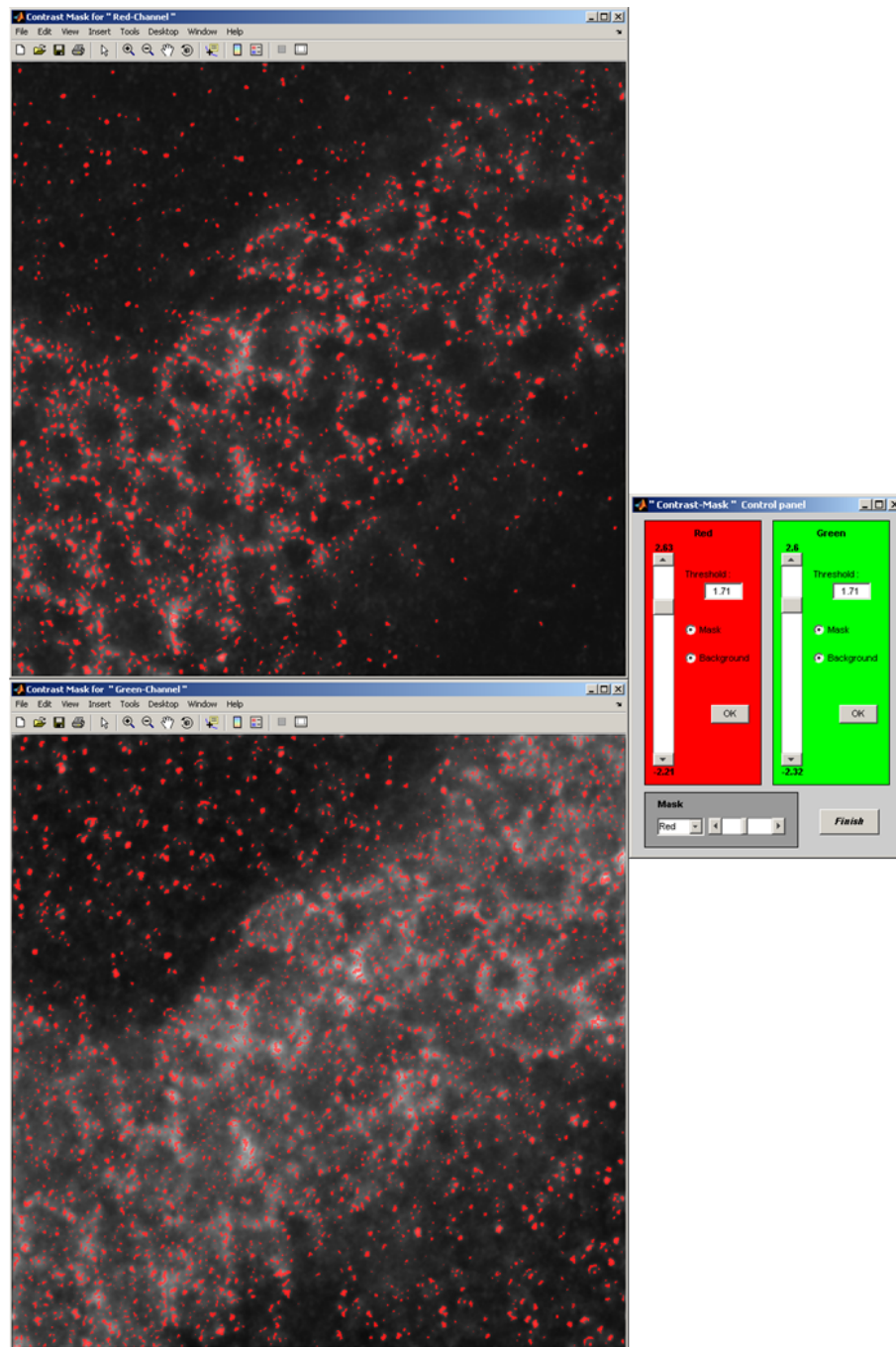


Figure B.2: Test2 (This GUI allows the user to set the threshold values for the modified contrast value of both images of the red-channel and green-channel, and hence define the contrast masks for the images. The background behind the contrast mask is the de-noised original image which is a reference for the user while selecting the threshold.)

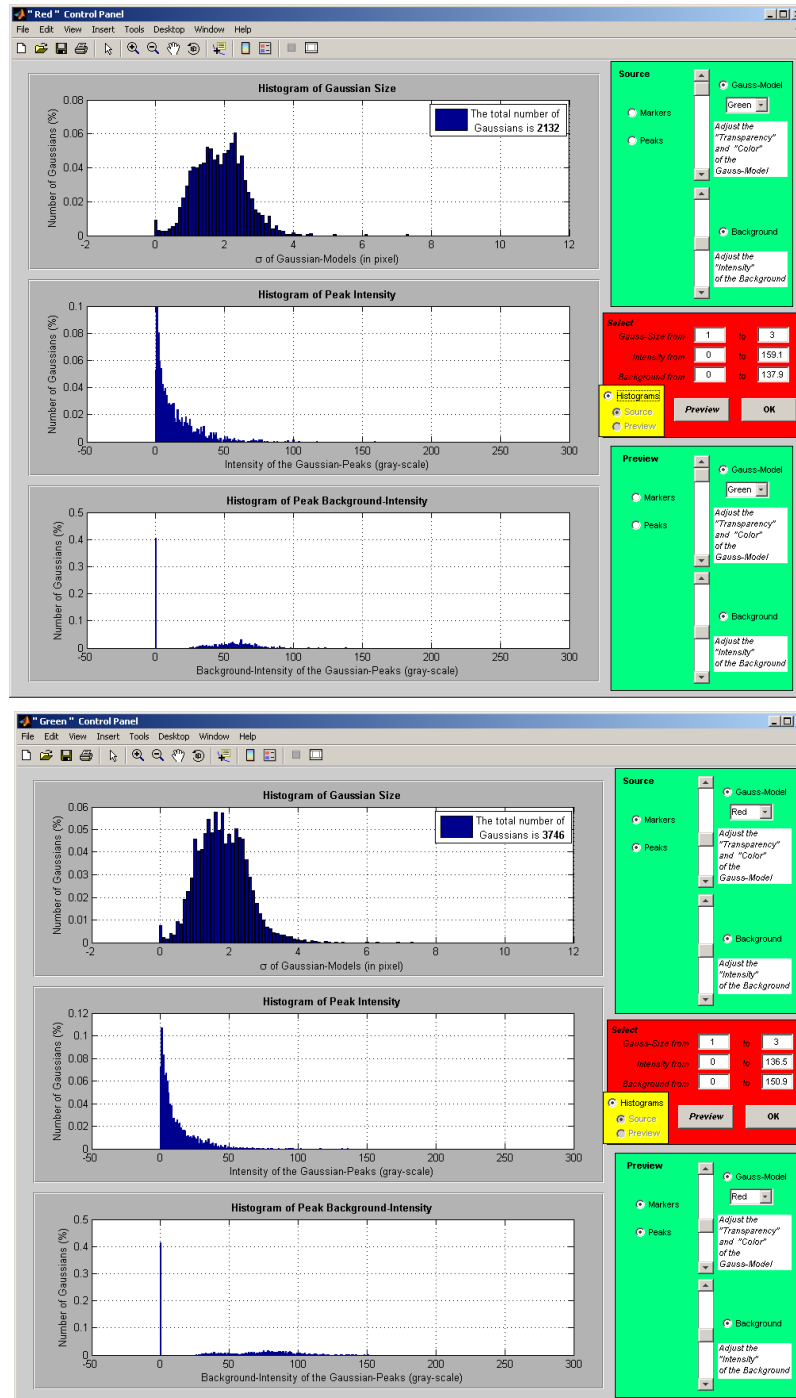


Figure B.3: Test3 I (This the first feature of third GUI, which displays the analytical histograms that mentioned in Chapter 4. It is also a reference for the user when setting threshold values for filtering the particle models.)

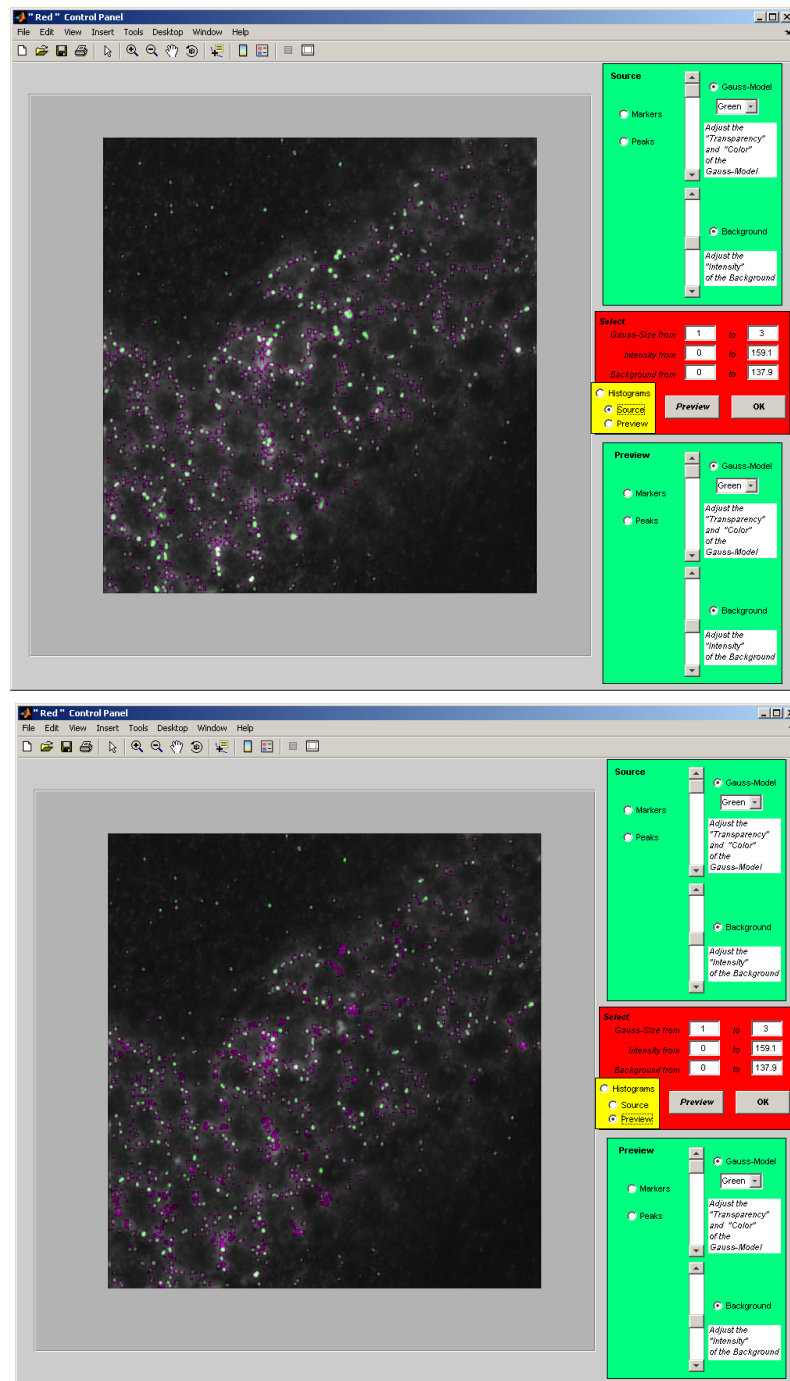


Figure B.4: Test3 II (This the second feature of third GUI, which displays the synthesized image of the estimate spot models. The de-noised original image can be turned on as the background of the image as a reference for the user when comparing to the estimated result. Also, some markers can be turned on to indicate the position and sizes of the estimated Gaussian components on the image.)

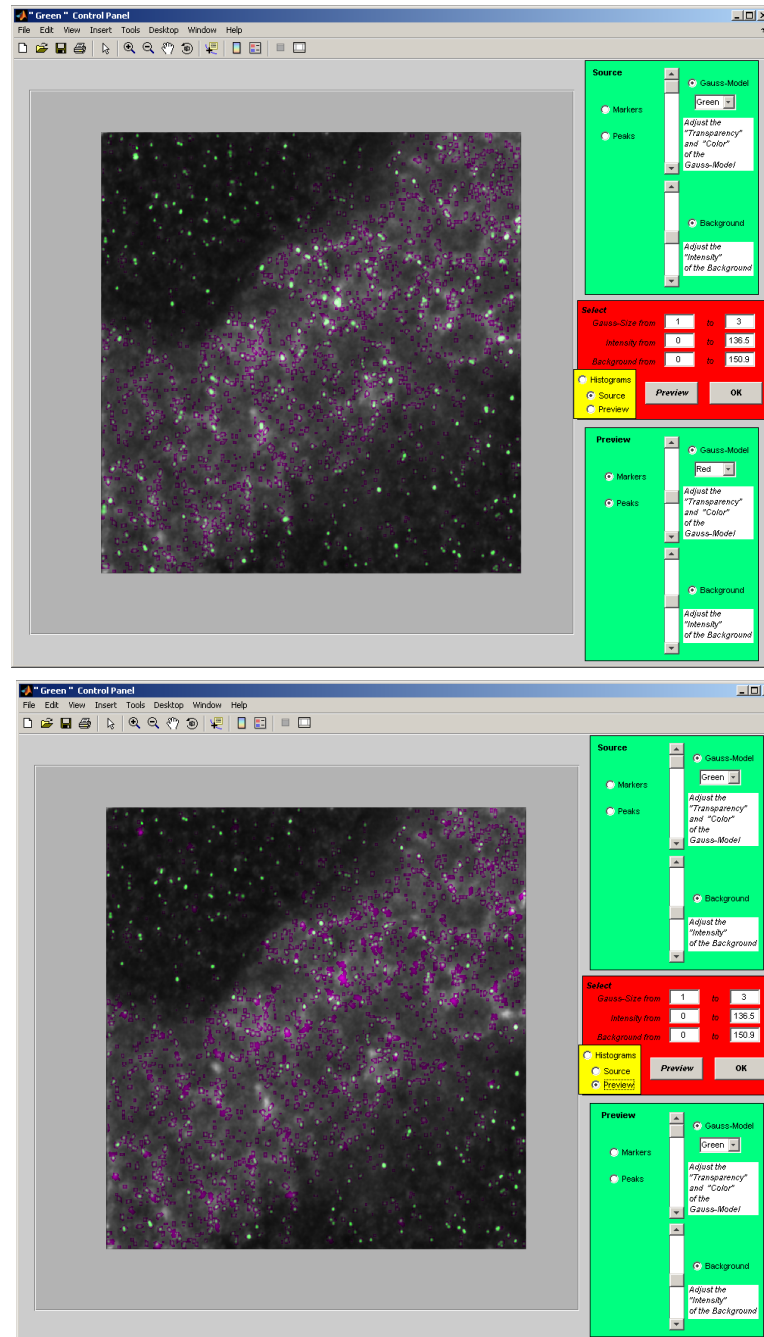


Figure B.5: Test3 III (This the third feature of third GUI, which allows the user to preview the thresholded image estimate when setting threshold values for filtering the particle models.)

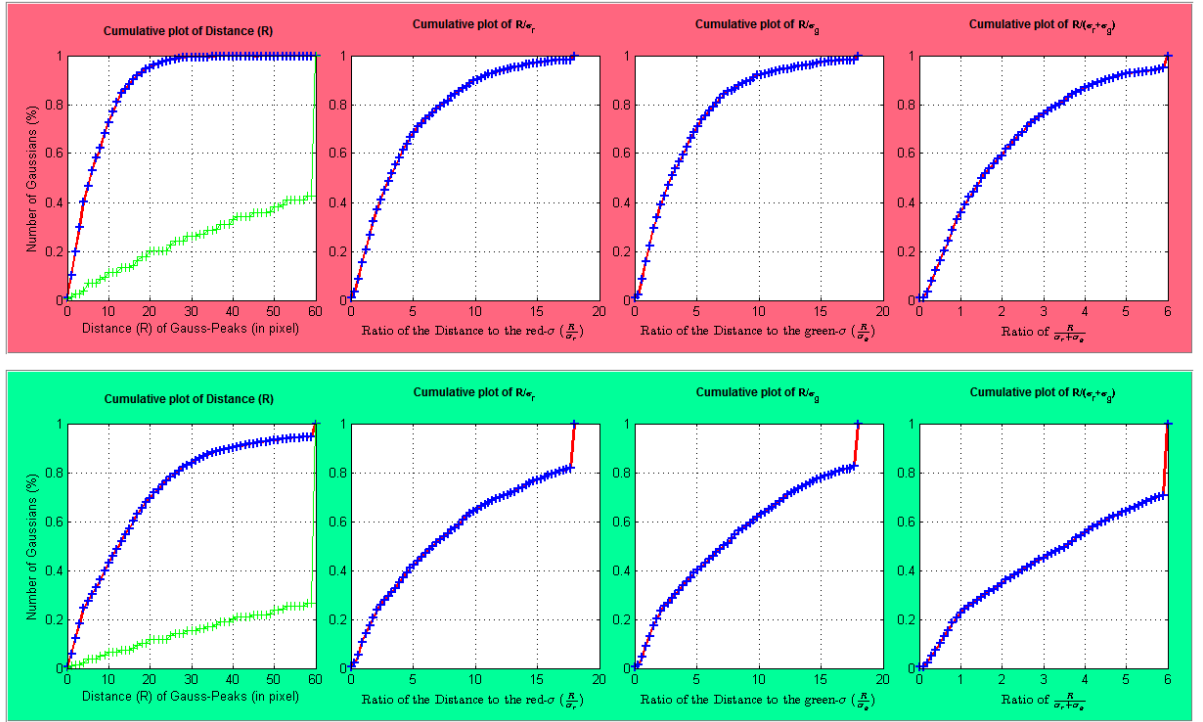


Figure B.6: Test3 IV (This the fourth feature of third GUI, which displays the cumulative plots as mentioned previously for co-localization analysis, where the top row and bottom row are respectively corresponding to the red-channel and green channel of a given image. The first column is, similar to Fig. 4.26(a)&(b), the percentage versus distance. The second column is the percentage versus the ratio of the distance to the size (σ) of the reference spot. The third column is the percentage versus the ratio of the distance to the size (σ) of the nearest neighbor spot in the other channel. The forth column is the percentage versus the ratio of the distance to the sum of the sizes (σ) of the reference and neighbor spots.)

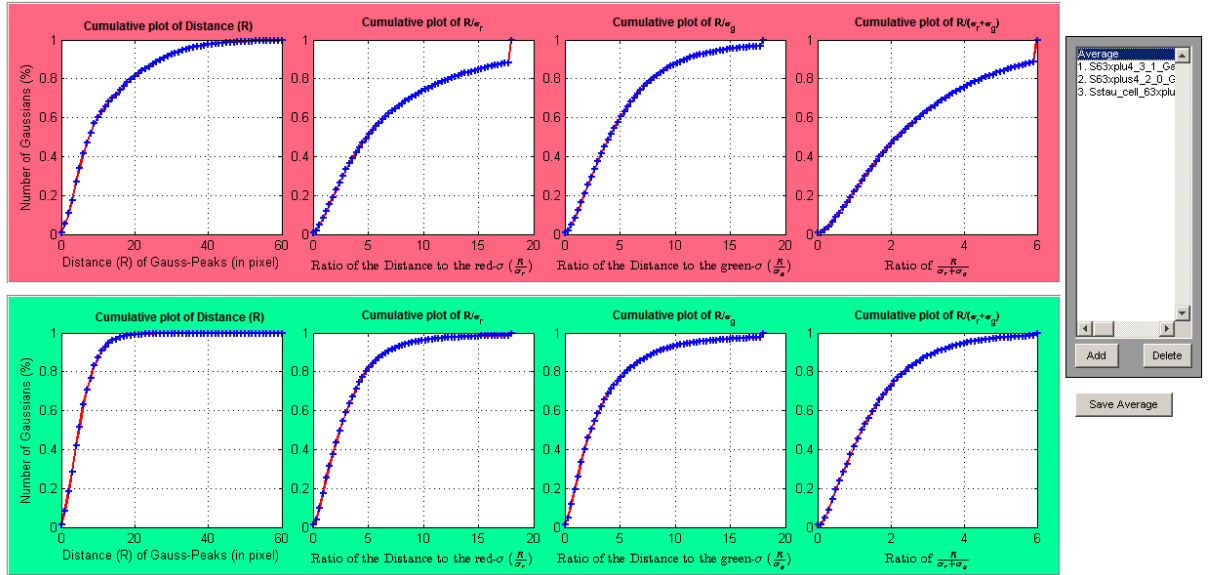


Figure B.7: Test4 (*This is the last GUI for averaging the co-localization analysis results of several sample images.*)

Bibliography

- [1] J. Angulo. Nucleus modelling and segmentation in cell clusters. In *Progress in Industrial Mathematics at ECMI'2008*,, pages 217–222, 2008.
- [2] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies. Image coding using wavelet transform. *Image Processing, IEEE Transactions on*, 1(2):205 –220, apr 1992.
- [3] X. Bai, C. Sun, and F. Zhou. Splitting touching cells based on concave points and ellipse fitting. *Pattern Recogn.*, 42(11):2434–2446, 2009.
- [4] P. Bamford and B. Lovell. Unsupervised cell nucleus segmentation with active contours. *Signal Process.*, 71(2):203–213, 1998.
- [5] A. Baradarani and R. Yu. A dual-tree complex wavelet with application in image denoising. *Proc. of ICIP 2007*, pages 1203–1206, 2007.
- [6] S. A. Barbee, P. S. Estes, A.-M. Cziko, J. Hillebrand, R. A. Luedeman, J. M. Coller, N. Johnson, I. C. Howlett, C. Geng, R. Ueda, A. H. Brand, S. F. Newbury, J. E. Wilhelm, R. B. Levine, A. Nakamura, R. Parker, and M. Ramaswami. Staufen- and fmrp-containing neuronal rnps are structurally and functionally related to somatic p bodies. *Neuron*, 52(6):997–1009, 2006.
- [7] N. P. Barrera, H. Ge, R. M. Henderson, W. J. Fitzgerald, and J. M. Edwardson. Automated analysis of the architecture of receptors, imaged by atomic force microscopy. *Micron*, 39(2):101–110, 2008.
- [8] G. Begelman, E. Gur, E. Rivlin, M. Rudzsky, and Z. Zalevsky. Cell nuclei segmentation using fuzzy logic engine. In *ICIP04*, pages 2937–2940, 2004.
- [9] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 48(3):259–302, 1986.
- [10] J. A. Bilmes. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical report, International Computer Science Institute, Berkeley CA & Computer Science Division, Department of Electrical Engineering and Computer Science, U.C. Berkeley, 1998.

- [11] M. V. Boland and R. F. Murphy. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of hela cells. *Bioinformatics*, 17(12):1213–1223, 2001.
- [12] S. Borman. The expectation maximization algorithm: A short tutorial. unpublished paper available at <http://www.seanborman.com/publications>. Introduces the Expectation Maximization (EM) algorithm and fleshes out the basic mathematical results, including a proof of convergence. The Generalized EM algorithm is also introduced., 2004.
- [13] A.-O. Boudraa and E.-H. S. Diop. Image contrast enhancement based on 2d teager-kaiser operator. *15th IEEE International Conference on Image Processing, 2008. ICIP 2008.*, pages 3180–3183, 2008.
- [14] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2001.
- [15] Y. Boykov and O. Veksler. Graph cuts in vision and graphics: Theories and applications. In N. Paragios, Y. Chen, and O. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 79–96. Springer US, New York, 2006.
- [16] C. R. Bramham and D. G. Wells. Dendritic mrna: transport, translation and function. *Nature reviews. Neuroscience*, 8(10):776–789, 2007.
- [17] L. Bretzner and T. Lindeberg. Feature tracking with automatic selection of spatial scales. *Computer Vision and Image Understanding*, 71(3):385 – 392, 1998.
- [18] J. H. Broeke, H. Ge, I. M. Dijkstra, A. T. Cemgil, J. A. Riedla, L. N. Cornelisse, R. F. Toonen, M. Verhage, and W. J. Fitzgerald. Automated quantification of cellular traffic in living cells. *Journal of Neuroscience Methods*, 178(2):378–384, 2009.
- [19] J. Buhler, T. Ideker, and D. Haynor. Dapple: improved techniques for finding spots on dna microarrays. *UW CSE Technical Report UWTP*, 2000.
- [20] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *Communications, IEEE Transactions on*, 31(4):532 – 540, apr 1983.
- [21] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Proc. of 5th ICCV*, pages 694–699, 1995.
- [22] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *Int. J. Comput. Vision*, 22(1):61–79, Feb. 1997.

- [23] T. Chan and L. Vese. An active contour model without edges. In *Proceedings of the Second International Conference on Scale-Space Theories in Computer Vision*, SCALE-SPACE '99, pages 141–151. Springer-Verlag, 1999.
- [24] J. Cheng and J. Rajapakse. Segmentation of clustered nuclei with shape markers and marking function. *IEEE Transactions on Biomedical Engineering*, 56(3):741–748, 2009.
- [25] L. D. Cohen. On active contour models and balloons. *CVGIP: Image Underst.*, 53(2):211–218, Mar. 1991.
- [26] S. Colantonio, O. Salvetti, and I. Gurevich. A two-step approach for automatic microscopic image segmentation using fuzzy clustering and neural discrimination. *Pattern Recognition and Image Analysis*, 17:428–437, 2007.
- [27] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:603–619, May 2002.
- [28] G. Cong and B. Parvin. Model-based segmentation of nuclei. *Pattern Recognition*, 33(8):1383 – 1393, 2000.
- [29] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.
- [30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.
- [31] O. Daněš, P. Matula, C. Ortiz-De-Solórzano, A. Muñoz Barrutia, M. Maška, and M. Kozubek. Segmentation of touching cell nuclei using a two-stage graph cut model. In *Proceedings of the 16th Scandinavian Conference on Image Analysis*, SCIA '09, pages 410–419. Springer-Verlag, 2009.
- [32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [33] M. Do and M. Vetterli. Framing pyramids. *Signal Processing, IEEE Transactions on*, 51(9):2329–2342, 2003.
- [34] R. Duda, P. Hart, and D. Stork. Pattern classification. *J. Classif.*, 24(2):305–307, 2007.
- [35] R. Farjam, H. Soltanian-Zadeh, K. Jafari-Khouzani, and R. A. Zoroofi. An image analysis approach for automatic malignancy determination of prostate pathological images. *Cytometry Part B: Clinical Cytometry*, 72B(4):227–240, 2007.

- [36] M. Figueiredo. Bayesian image segmentation using gaussian field priors. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, volume 3757 of *Lecture Notes in Computer Science*, pages 74–89. Springer Berlin Heidelberg, 2005.
- [37] M. Flierl and P. Vanderghelynst. An improved pyramid for spatially scalable video coding. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II–878–81, 2005.
- [38] A. Foulonneau, P. Charbonnier, and F. Heitz. Multi-reference shape priors for active contours. *Int. J. Comput. Vision*, 81(1):68–81, 2009.
- [39] D. Gil and P. Radeva. Curvature vector flow to assure convergent deformable models for shape modelling. In *EMMCVPR*, volume 2683 of *Lecture Notes in Computer Science*, pages 357–372. Springer, 2003.
- [40] K. J. Gilmore, M. Kita, Y. Han, A. Gelmi, M. J. Higgins, S. E. Moulton, G. M. Clark, R. Kapsa, and G. G. Wallace. Skeletal muscle cell proliferation and differentiation on polypyrrole substrates doped with extracellular matrix components. *Biomaterials*, 30:589–601, 2009.
- [41] P. C. Goodwin. Evaluating optical aberration using fluorescent microspheres: methods, analysis, and corrective actions. *Methods Cell Biol*, 81:397–413, 2007.
- [42] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Royal Journal on Statistical Society*, 51(2):271–279, 1989.
- [43] R. Haralick and L. Shapiro. *Computer and robot vision*. Number v. 2 in *Computer and Robot Vision*. Addison-Wesley Pub. Co., 1993.
- [44] D. J. Heeger and J. R. Bergen. Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, SIGGRAPH '95, pages 229–238, 1995.
- [45] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 453–460, New York, NY, USA, 1998. ACM.
- [46] J. Hillebrand, K. Pan, A. Kokaram, S. Barbee, R. Parker, and M. Ramaswami. The me31b dead-box helicase localizes to postsynaptic foci and regulates expression of a camkii reporter mrna in dendrites of drosophila olfactory projection neurons. *Frontiers in Neural Circuits*, 4(00121), 2010.
- [47] M. Hilton, B. D. Jawerth, and A. Sengupta. Compressing still and moving images with wavelets. *Multimedia Systems*, 2(3):218–227, 1994.

- [48] M. Hu, X. Ping, and Y. Ding. Automated cell nucleus segmentation using improved snake. In *Image Processing, 2004. ICIP '04. 2004 International Conference on*, volume 4, pages 2737 – 2740, 2004.
- [49] I. Izeddin, J. Boulanger, V. Racine, C. Specht, A. Kechkar, D. Nair, A. Triller, D. Choquet, M. Dahan, and J. Sibarita. Wavelet analysis for single molecule localization microscopy. *Opt. Express*, 20(3):2081–2095, 2012.
- [50] J. Jensen. Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, 30(1):175–193, 1906.
- [51] M. Jeong, B. Ko, and J. Nam. Overlapping nuclei segmentation based on bayesian networks and stepwise merging strategy. *Journal of Microscopy*, 235(2):188–198, 2009.
- [52] C. Jung and C. Kim. Segmenting clustered nuclei using h-minima transform-based marker extraction and contour parameterization. *Biomedical Engineering, IEEE Transactions on*, 57(10):2600–2604, 2010.
- [53] J. F. Kaiser. On a simple algorithm to calculate the 'energy' of a signal. *Proc. of ICASSP 1990*, 1:381–384, 1990.
- [54] M. Kass, A. Witkin, and D. Terzopoulos. Active contour models. *Proc. of 1st ICCV*, pages 259–268, 1987.
- [55] J. Kittler and J. Illingworth. Minimum error thresholding. *Pattern Recogn.*, 19(1):41–47, Jan. 1986.
- [56] J. Kovacevic and A. Chebira. Life beyond bases: The advent of frames. *Signal Processing Magazine, IEEE*, 24(4):86–104, 2007.
- [57] G. Lebrun, C. Charrier, O. Lezoray, C. Meurie, and C. H. A fast and efficient segmentation scheme for cell microscopic image. *Cellular and molecular biology*, 53(2):51–61, 2007.
- [58] B. Li and S. T. Acton. Active contour external force using vector field convolution for image segmentation. *Trans. Img. Proc.*, 16(8):2096–2106, Aug. 2007.
- [59] X. Li, M. Hao, D. W. Piston, and B. M. Dawant. Automatic tracking of proteins in sequences of fluorescence images. *Medical Imaging 2004: Image Processing. Edited by Fitzpatrick, J. Michael; Sonka, Milan. Proceedings of the SPIE*, 5370:1364–1371, 2004.
- [60] Y. Li and L. Li. A novel split and merge em algorithm for gaussian mixture model. In *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, volume 6, pages 479 –483, aug. 2009.

- [61] L. Liu, L. Gan, and T. Tran. General reconstruction of laplacian pyramid and its dual frame solutions. In *Information Sciences and Systems, 2007. CISS '07. 41st Annual Conference on*, pages 514–517, 2007.
- [62] P. Loo and N. Kingsbury. Watermarking using complex wavelets with resistance to geometric distortion. *10th European Signal processing Conference (EUSIPCO 2000)*, pages 2543–2560, 2000.
- [63] J. Ma and Q. He. A dynamic merge-or-split learning algorithm on gaussian mixture for automated model selection. *Intelligent data engineering and automated learning, Brisbane, Australia*, 3578(6):203–210, 2005.
- [64] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980.
- [65] M. K. Mihçak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients, 1999.
- [66] M. Minsky. Memoir on inventing the confocal scanning microscope. *Scanning*, 10:128–138, 1988.
- [67] S. K. Mitra, H. Li, I. song Lin, and T. hu Yu. A new class of nonlinear filters for image enhancement. *International Conference on Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.*, 4:2525–2528, 1991.
- [68] K. Mosaliganti, A. Gelas, A. Gouaillard, and S. Megason. Tissue level segmentation and tracking of biological structures in microscopic images based on density maps. In *Proceedings of the 6th IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, ISBI'09, pages 1354–1357. IEEE Press, 2009.
- [69] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989.
- [70] K. P. Murphy. *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*. The MIT Press, 2012.
- [71] S. Naik, S. Doyle, S. Agner, A. Madabhushi, M. Feldman, and J. Tomaszewski. Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology. In *ISBI*, pages 284–287. IEEE, 2008.
- [72] S. Naik, S. Doyle, M. Feldman, J. Tomaszewski, and A. Madabhushi. Gland segmentation and computerized gleason grading of prostate histology by integrating low-, high-level and domain specific information. In *2nd MICCAI Workshop Microscopic Image Analysis with Appl. in Biology*, 2007.

- [73] R. Nakano, Z. Ghahramani, G. E. Hinton, and N. Ueda. Smem algorithm for mixture models. *Neural Computation*, 12(9):2109–2128, 2000.
- [74] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, Nov. 1988.
- [75] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(1):62–66, 1979.
- [76] K. Pan, D. Corrigan, J. Hillebrand, M. Ramaswami, and A. Kokaram. A wavelet-based bayesian framework for 3d object segmentation in microscopy. In *SPIE BiOS Symposium*, San Francisco, United State, 2012.
- [77] K. Pan, J. Hillebrand, M. Ramaswami, and A. Kokaram. Gaussian mixture models for spots in microscopy using a new split/merge em algorithm. In *IEEE International Conference on Image Processing (ICIP’10)*, pages 3645 – 3648, Hong Kong, China, October 2010.
- [78] K. Pan, J. Hillebrand, M. Ramaswami, and A. Kokaram. Gaussian mixtures for intensity modeling of spots in microscopy. In *International Symposium on Biomedical Imaging (ISBI’10)*, pages 121 – 124, Rotterdam, Netherlands, February 2010.
- [79] K. Pan, A. C. Kokaram, K. Gilmore, M. J. Higgins, R. Kapsa, and G. G. Wallace. Cell-snake: A new active contour technique for cell/fibre segmentation. In *IEEE International Conference on Image Processing (ICIP’11)*, pages 2153–2156, Brussels, Belgium, September 2011.
- [80] N. Paragios, O. Mellina-Gottardo, and V. Ramesh. Gradient vector flow fast geometric active contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(3):402–407, Mar. 2004.
- [81] J. Portilla and E. P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–71, 2000.
- [82] X. Qi, F. Xing, D. Foran, and L. Yang. Robust segmentation of overlapping cells in histopathology specimens using parallel seed detection and repulsive level set. *Biomedical Engineering, IEEE Transactions on*, 59(3):754 –765, march 2012.
- [83] T. W. Ridler and S. Calvard. Picture thresholding using an iterative selection method. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(8):630–632, 1978.
- [84] R. Russell, N. Adams, D. Stephens, E. Batty, J. K., and P. Freemont. Segmentation of fluorescence microscopy images for quantitative analysis of cell nuclear architecture. *Biophysical Journal*, 96(8):3379–3389, 2009.
- [85] P. Sarder, A. Nehorai, P. H. Davis, and S. L. Stanley. Estimating gene signals from noisy microarray images. *IEEE Transactions on NanoBioscience*, 7(2):142–153, 2008.

- [86] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury. The dual-tree complex wavelet transform. *IEEE Transactions on Signal Processing*, 22:123–151, 2005.
- [87] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005.
- [88] P. Soille. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2 edition, 2003.
- [89] H. M. Teager. Evidence for nonlinear sound production mechanisms in the vocal tract. In W. J. Hardcastle and A. Marchal, editors, *Speech Production and Speech Modelling*, pages 241–261. Kluwer Academic, Boston, MA, 1990.
- [90] L. Vincent. Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. *IEEE Transactions on Image Processing*, 2:176–201, 1993.
- [91] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, and S. T. Wong. Novel cell segmentation and online svm for cell cycle phase identification in automated microscopy. *Bioinformatics*, 24(1):94–101, 2008.
- [92] G. Xiong, C. Feng, and L. Ji. Dynamical gaussian mixture model for tracking elliptical living objects. *Pattern Recognition Letters*, 27(7):838–842, 2006.
- [93] C. Xu and J. L. Prince. Generalized gradient vector flow external forces for active contours. *Signal Process.*, 71(2):131–139, Dec. 1998.
- [94] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *Trans. Img. Proc.*, 7(3):359–369, Mar. 1998.
- [95] H.-F. Yang and Y. Choe. Cell tracking and segmentation in electron microscopy images using graph cuts. In *Biomedical Imaging: From Nano to Macro, 2009. ISBI '09. IEEE International Symposium on*, pages 306–309, 2009.
- [96] L. Yang, P. Meer, and D. J. Foran. Unsupervised segmentation based on robust estimation and color active contour models. *IEEE Trans. on Information Technology in Biomedicine*, 9:475–486, 2005.
- [97] B. Zhang, J. Zerubia, and J.-C. Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Applied Optics*, 46(10):1819–29, 2007.
- [98] B. Zhang, J. Zerubia, and J.-C. Olivo-Marin. Gaussian approximations of fluorescence microscope point-spread function models. *Appl. Opt.*, 46(10), 2007.
- [99] Z. Zhang, C. Chen, J. Sun, and K. L. Chan. Em algorithms for gaussian mixtures with split-and-merge operation. *Pattern Recognition*, 36(9):1973–1983, 2003.