

**PERCEPTUAL WATERSHEDS FOR CELL
SEGMENTATION IN FLUORESCENCE
MICROSCOPY IMAGES**

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Salim Arslan

August, 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. ıgdem Gündüz Demir(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Rengül Çetin Atalay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Pınar Duygulu Şahin

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

PERCEPTUAL WATERSHEDS FOR CELL SEGMENTATION IN FLUORESCENCE MICROSCOPY IMAGES

Salim Arslan

M.S. in Computer Engineering

Supervisor: Assist. Prof. Dr. Çiğdem Gündüz Demir

August, 2012

High content screening aims to analyze complex biological systems and collect quantitative data via automated microscopy imaging to improve the quality of molecular cellular biology research in means of speed and accuracy. More rapid and accurate high-throughput screening becomes possible with advances in automated microscopy image analysis, for which cell segmentation commonly constitutes the core step. Since the performance of cell segmentation directly affects the output of the system, it is of great importance to develop effective segmentation algorithms. Although there exist several promising methods for segmenting monolayer isolated and less confluent cells, it still remains an open problem to segment more confluent cells that grow in aggregates on layers.

In order to address this problem, we propose a new marker-controlled watershed algorithm that incorporates human perception into segmentation. This incorporation is in the form of how a human locates a cell by identifying its correct boundaries and piecing these boundaries together to form the cell. For this purpose, our proposed watershed algorithm defines four different types of primitives to represent different types of boundaries (left, right, top, and bottom) and constructs an attributed relational graph on these primitives to represent their spatial relations. Then, it reduces the marker identification problem to the problem of finding predefined structural patterns in the constructed graph. Moreover, it makes use of the boundary primitives to guide the flooding process in the watershed algorithm. Working with fluorescence microscopy images, our experiments demonstrate that the proposed algorithm results in locating better markers and obtaining better cell boundaries for both less and more confluent cells, compared to previous cell segmentation algorithms.

Keywords: Cell segmentation, fluorescence microscopy imaging, marker-controlled watershed, watershed, attributed relational graphs.

ÖZET

FLORESAN MİKROSKOP GÖRÜNTÜLERİNDE HÜCRE BÖLÜTLEMESİ İÇİN ALGISAL SU-SEDDİ ALGORİTMASI

Salim Arslan

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Yrd. Doç. Dr. Çiğdem Gündüz Demir

Ağustos, 2012

Yüksek işlem hacimli içerik taraması, floresan mikroskop görüntüleri kullanarak, karmaşık biyolojik sistemlerin yüksek hız ve başarı oranıyla analizine ve sayısal veri elde edilmesine olanak sağlar; böylelikle, moleküler hücresel biyoloji araştırmalarının kalitesinin artırılması hedeflenir. Daha hızlı ve daha hatasız tarama, otomatik mikroskopik görüntü analizi sistemlerindeki gelişmelerle mümkündür. Bu sistemlerde genellikle ana adım, görüntülerdeki hücrelerin doğru bir şekilde bölütlenmesidir. Bölütleme işleminin sonuçları, sistemin sonraki adımlarını doğrudan etkileyeceğinden, verimli bölütleme algoritmaları geliştirmek büyük bir önem taşımaktadır. Literatürde, tekil ve az kalabalık hücrelerden oluşan görüntüleri bölütlemek üzere tasarlanmış umut verici yöntemler olsa da, üst üste büyüyen, daha kalabalık hücreleri bölütlemek halen çözüm bekleyen bir problem olarak yerini korumaktadır.

Bu tezde, bu problemi çözmek üzere, insan algısını hücre bölütleme ile bağdaştıran yeni bir işaretçi-kontrollü su-seddi algoritması sunulmaktadır. Bu bağlamda, bir insanın bir hücrenin doğru kenarlarını algılayıp, bunları bir araya getirmek suretiyle hücrenin yerini saptaması, hücre bölütleme probleminin çözümüne ilham kaynağı olmuştur. Bu amaçla sunulan su-seddi algoritması, farklı tipteki kenarları (sol, sağ, üst ve alt) temsil eden dört farklı tipte primitif tanımlar ve bir özellikli ilişkisel çizge ile primitiflerin birbirleriyle olan konumsal ilişkilerini modeller. Böylece işaretçi bulma problemi, çizge içerisinde önceden tanımlanmış yapısal örüntüleri arama problemine indirgenmiş olur. Ayrıca geliştirilen yöntem, kenar primitiflerinden faydalanarak su-seddi algoritmasında suyun akışını kontrol eder. Floresan görüntüler üzerinde yapılan deneyler, sunulan algoritmanın,

hem az kalabalık hem de çok kalabalık hücre görüntülerinde, önceki algoritmalara kıyasla işaretçileri daha iyi tanımladığını ve hücreleri daha iyi bölütlediğini göstermiştir.

Anahtar sözcükler: Hücre bölütleme, floresan mikroskopik görüntüleme, işaretçikontrollü su-seddi, su-seddi, özellikli ilişkisel çizge.

Acknowledgement

I would like to express my gratitude to my supervisor Assist. Prof. Dr. ıgdem Gündüz Demir for her support and guidance throughout this thesis study, which would not be existed without her invaluable assistance and supervision. I have learned a lot from her about both computer science and how to become a good supervisor.

I would like to express my great appreciation to the members of my supervisory committee, Assist. Prof. Dr. Pınar Duygulu Şahin and Assoc. Prof. Dr. Rengül Çetin Atalay for their time to evaluate this thesis. I wish to acknowledge the help provided by Tülin Erşahin and İrem Durmaz for acquisition of the data and relevant information.

I would like to thank to my undergraduate supervisor Assoc. Prof. Dr. Aybars Uğur, who introduced me the way of doing science and encouraged me for a career in academia.

I would like to convey thanks to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing financial assistance during my graduate studies.

I would like to offer my special thanks to my graduate friends, especially, Erdem, Burak, Barış, Mustafa, Zeynep, Oğuz, Erdem, Merve, Çağdaş, Alper, and Shatlyk for their friendship and support, as well as for making time in the office more sufferable. My special thanks are extended to Can, who has been a valuable research partner, Onur, who made the office more enjoyable with his delicious taste in music, and Necip, who has been a perfect companion for the last two years.

Last but not least, I would like to express my love and gratitude to my family and to my beloved wife, Dilara, for their understanding and endless love, through the duration of my graduate studies. Anything would not be possible without their help, patience and support.

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Contribution	5
1.3	Outline	6
2	Background	7
2.1	Domain Description	7
2.1.1	Fluorescence Microscopy Images	7
2.1.2	Cell Lines	10
2.2	High Content Screening	11
2.2.1	Image Preprocessing	13
2.2.2	Cell/Nucleus Segmentation	14
3	Methodology	25
3.1	Primitive Definition	25
3.2	Marker Identification	29

3.2.1	Graph Construction	30
3.2.2	Iterative Search Algorithm	31
3.2.3	Cell Localization	32
3.3	Region Growing	35
4	Experiments	37
4.1	Dataset	37
4.2	Comparisons	38
4.3	Evaluation	39
4.4	Parameter Selection	40
4.5	Results	41
4.6	Parameter Analysis	45
5	Conclusion	53

List of Figures

1.1	Example subimages of differently grown cells: (a), (b) are monolayer/isolated cells that have no or very little contact with others; (c), (d) are touching cells that have an adjacent or a very close cell; (e), (f) are confluent cells which grow in aggregates and observed as overlapping objects in the image. The segmentation process gets more challenging from (a) to (f).	3
1.2	(a) An image of HepG2 hepatocellular carcinoma cell nuclei. (b) For four individual cell nuclei, the left, right, top, and bottom boundaries are shown as green, yellow, pink, and red, respectively.	4
2.1	Several example fluorescence microscope image segments: (a), (c), (e) are taken from the HepG2 dataset; (b), (d), (f) are taken from the Huh7 dataset. The images evidently share some features such as color, bright foreground and dark background, but also show some differences in texture and illumination due to different cell lines and technical problems emerged during the image acquisition.	9
2.2	High content screening pipeline, which consists of image acquisition, image preprocessing, cell/nucleus segmentation, cell tracking and registration, feature extraction, data modeling and storage, statistical analysis, and visualization.	12

2.3 Results of applying preprocessing methods to the image: (a) an example fluorescence image with high intensity variations and nonuniform shades due to uneven illumination, (b) the resulting image after preprocessing. 14

2.4 Illustrations of true, oversegmented, and undersegmented cells: (a) a sample fluorescence microscopy image, (b) segmentation results, where oversegmented cells are annotated with red-cyan lines, undersegmented cells are annotated with red dotted lines, and true cells are annotated with yellow lines, (c) segmentation delineated by an expert, where each color represents a cell. 15

2.5 The results of applying global and local thresholding to an example image: (a) a sample fluorescence microscopy image, (b) the segmentation result of global thresholding, (c) the segmentation result of local thresholding. Due to uneven illumination, global thresholding classifies relatively darker cellular pixels as background. 18

2.6 Illustration of how snakes work: (a) a sample fluorescence image, (b) initial points provided around object boundaries, (c) initial boundaries, (d) final segmentation results. When initial points are given properly, the spline delineates the object boundaries, but with some limitations: (i) the spline may not localize concave curves accurately, (ii) touching cells may not be separated, which yields undersegmentation. 18

2.7 Illustration of active contours without edges: (a) a sample fluorescence image, (b) the randomly assigned first spline, (c) splines after a few hundred iterations, (d) final segmentation results. Active contours without edges can converge into boundaries regardless of the initial points. Contrary to the snakes as illustrated in Fig. 2.6, it better manages to separate touching cells. 19

2.8 Illustration of a watershed in the field of image processing: (a) a synthetically generated gray scale image of two dark blobs, (b) 3D surface plot of the intensities where the colors of the points in the space turn into yellow as the intensity of the pixels increases. Starting from the minima (the darkest red), catchment basins merge onto the watershed line, illustrated with a dotted red line. 21

2.9 An example segmentation result obtained by a watershed algorithm applied to only gradient magnitudes: (a) a sample fluorescence image, (b) the gradient magnitudes, (c) 3D illustration of the gradient image, in which local minima can be observed around boundaries and the centroids of cells, (d) labeled segmented regions after applying watershed to the gradients. Since the flooding starts with all minima at the same time and catchment basins join as soon as they meet, the image is highly oversegmented. 22

2.10 An example to show that regional minima and suppressed minima cannot handle oversegmentation problem when intensities are uneven: (a) gray-scale intensity image, (b) minima of the intensity image, (c) segmentation results using regional minima as markers, (d) suppressed regional minima via h -minima transform, (e) segmentation results using suppressed minima as markers. Although h -minima transform is used, the oversegmentation problem may still exist. Note that, for better illustration, a mask was used to eliminate noise in the background. 23

2.11 Integrating shape information in a marker-controlled watershed via distance maps: (a) the gray scale intensity map of the image in Fig. 2.9, (b) binary map obtained via Otsu’s thresholding method [1], (c) the inner distance transform map where the pixels around cell centroids have the furthest distance to the background, (d) reverse of the inner distance transform map where the regional minima corresponds to the markers, (e) the segmentation result after applying marker-controlled watershed algorithm to the binary map. 23

3.1 Overview of the proposed algorithm. 26

3.2 Illustration of defining left boundary primitives: (a) original subimage, (b) response map R_{left} obtained by applying the Sobel operator of left orientation, (c) mask that is to be used for determining local Sobel threshold levels, (d) binary image B_{left} after thresholding, (e) boundaries obtained after taking the leftmost pixels, (f) boundary map P_{left} obtained after taking the d -leftmost pixels, (g) P_{left} after eliminating its smaller connected components, (h) left boundary primitives each of which is shown with a different color. 27

3.3 Illustration of the benefits of using a mask: (a) original subimage, (b) response map R_{bottom} obtained by applying the Sobel operator of bottom orientation, (c) bottom boundary primitives obtained without a mask (a falsely detected primitive is marked with red), (d) bottom boundary primitives obtained with a mask. 29

3.4 Illustration of assignment of an edge between a left and a bottom primitive: (a) primitives and (b) selected segments of the primitives. 30

3.5 Flowchart of the iterative search algorithm. 32

3.6 Two structural patterns used for cell localization: 4PRIM and 3PRIM patterns. Some instances of these patterns are shown in this figure, indicating the corresponding edges with black and blue, respectively. Moreover, the 4PRIM pattern has two subtypes that correspond to subgraphs forming a loop (dashed black edges) and a chain (solid black edges). 33

3.7 Flowchart of the cell localization method. 34

3.8 Illustration of finding outermost pixels and calculating a radial distance r_i . The primitive segments that lie in the correct sides of the other primitives are identified and outermost pixels are selected. In this figure, unselected segments of the primitives are indicated as gray. 35

3.9 (a) Original subimage, (b) primitives identified for different cells, and (c) cells delineated after the watershed algorithm. 36

4.1 Visual results on example subimages obtained by the proposed perceptual watershed algorithm and the comparison methods. The size of the subimages is scaled for better visualization. 42

4.2 For the Huh7 and HepG2 datasets, cell-based and pixel-based F-score measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element. . . 46

4.3 For the Huh7 and HepG2 datasets, cell-based and pixel-based precision measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element. . . 47

4.4	For the Huh7 and HepG2 datasets, cell-based and pixel-based recall measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element.	48
4.5	Visual results on example subimages obtained by the proposed perceptual watershed algorithm.	49
4.6	Visual results on example subimages obtained by the proposed perceptual watershed algorithm.	50
4.7	Visual results on example subimages obtained by the proposed perceptual watershed algorithm.	51
4.8	Visual results on example subimages obtained by the proposed perceptual watershed algorithm.	52

List of Tables

4.1	Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of computed-annotated cell matches. The results are obtained on the Huh7 dataset.	43
4.2	Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of computed-annotated cell matches. The results are obtained on the HepG2 dataset.	43
4.3	Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the cell-based precision, recall, and F-score measures. The results are obtained on the Huh7 dataset.	44
4.4	Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the cell-based precision, recall, and F-score measures. The results are obtained on the HepG2 dataset.	44
4.5	Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the pixel-based precision, recall, and F-score measures. The results are obtained on the Huh7 dataset.	44

4.6 Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the pixel-based precision, recall, and F-score measures. The results are obtained on the HepG2 dataset.	45
--	----

Chapter 1

Introduction

High content screening helps scientists analyze complex biological systems and collect quantitative data via automated microscopy imaging to improve the quality of molecular biology research in means of speed and accuracy. In the last two decades, automated fluorescence microscopy imaging systems have become important tools, particularly in high content screening and drug discovery as they enable to carry out rapid high-throughput screening with better reproducibility. The first step of these systems typically includes cell/nucleus segmentation. It is the most critical part in cellular image analysis, since it greatly affects the success of the other system steps. Thus, it is of great importance to develop accurate segmentation algorithms, considering the requirements of a given problem. In different applications of the biology research, different types of cells that show different characteristics can be used. For example in drug discovery screening, it is essential to compare drug-treated cells with non-treated control cells for driving reliable assessment on the cytotoxic effects of the drug. The drug-treated cells mostly grow as monolayer isolated cells whereas the drug-free control cells usually grow in aggregates on layers, which makes them more confluent. Thus, it is very critical for segmentation algorithms developed for drug discovery screening to operate on both isolated and confluent cells. Therefore, the proposed study in this thesis aims to develop such a cell segmentation method, which is capable of segmenting isolated and confluent cells.

1.1 Motivation

Standard drug discovery techniques require identifying the active compound by traditional methods, however, since molecular secrets about diseases have been revealed, it is now possible to find the compound that is responsible for the disease at molecular level. High throughput screening allows quantitative analysis in drug discovery by making possible to conduct thousands of experiments in a considerable amount of time. Especially the 2000s has been a golden era in the field, where efficient and effective methods have been developed in both academia and industry. Owing to increasing demands for newer and better applications, the research in the field is being evolved in a wide range and new solutions have been proposed both in software and hardware [2]. However, the tools to process and analyze the data are far away from perfection and they have still a lot of way to fulfill all the requirements in the field.

Although high content screening includes several steps to produce quantitative results, one of the major steps is the detection and segmentation of cells. Since the aim of the screening in drug discovery is to identify how drugs affect the phenotype of a cell, the importance of segmenting cells with a high accuracy is obvious. Therefore, in this study, we focus on cell detection and segmentation in fluorescence microscopy images.

In literature, there have been many studies proposed for cell/nucleus segmentation. These studies typically consider the specific characteristics of fluorescence microscopy images, such as sharp intensity changes between cell nuclei and the background, to develop their algorithms. When the images mostly consist of monolayer isolated or less confluent cells, relatively simple methods such as thresholding [3, 4] are used to separate cell nuclei from the background. On the other hand, these methods are typically inadequate for segmenting more confluent cells that grow in aggregates. In that case, it has been proposed to use watershed algorithms that operate on the intensity/gradient of image pixels and/or the shape information derived from a binary mask of the image [5, 6]. A typical problem of the watershed algorithms is over-segmentation. Marker-controlled watersheds, which define a set of markers and let water rise only from these markers,

have been used to overcome this problem [7, 8, 9]. Moreover, the watershed algorithms usually refine their results by applying a merging or a splitting process to their segmented cells. They split or merge the segmented cells based on the properties of their regions as well as the similarity between the adjacent cells and their boundaries [10, 11].

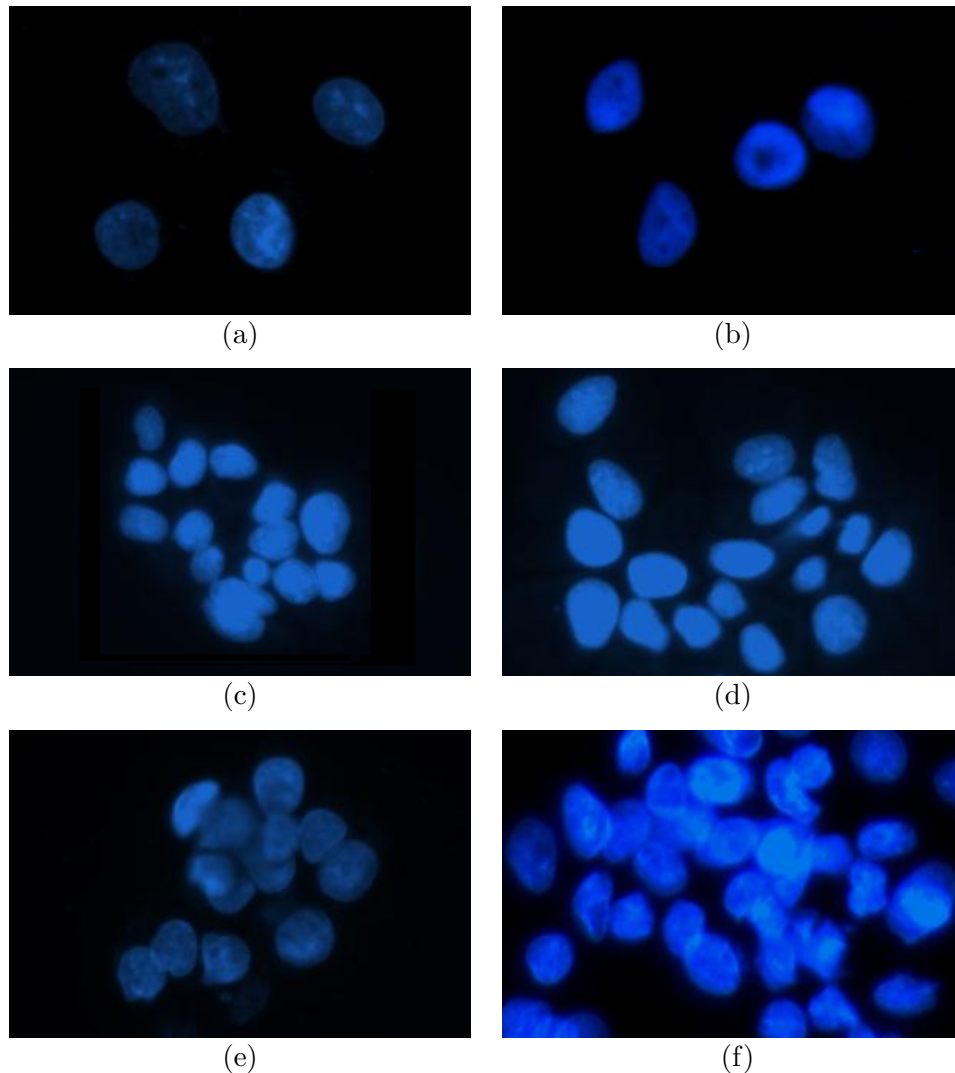


Figure 1.1: Example subimages of differently grown cells: (a), (b) are mono-layer/isolated cells that have no or very little contact with others; (c), (d) are touching cells that have an adjacent or a very close cell; (e), (f) are confluent cells which grow in aggregates and observed as overlapping objects in the image. The segmentation process gets more challenging from (a) to (f).

Although these previous studies lead to promising results, there still remain challenges to overcome for especially segmentation of confluent (clustered) cells. To make the isolated and confluent cells concepts more clear, we present some example images of cell groups in Fig. 1.1. The figure reveals that it is relatively easier to segment isolated (a, b) and touching cells (c, d). On the other hand, segmentation of confluent cells (e, f) needs a great effort, but not usually for human eyes. The main challenge in segmenting objects actually lies in the nature of the problem. Cell segmentation, like all other segmentation problems, is closely related with human perception. Humans typically use their perceptions in handling noise and variations in an image as well as in separating confluent cells from each other.

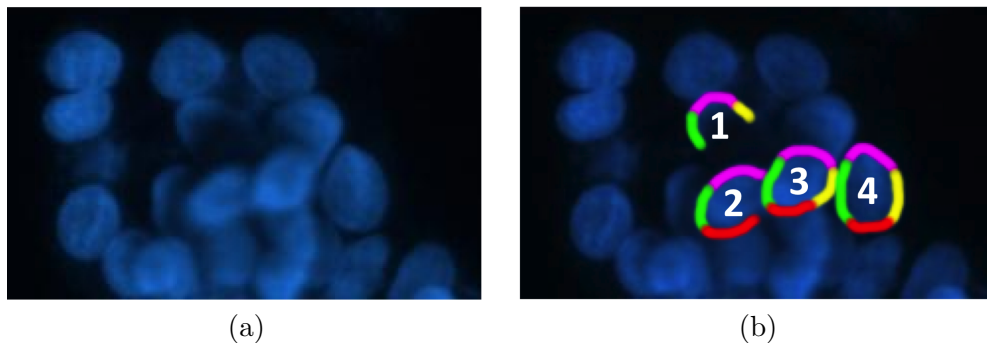


Figure 1.2: (a) An image of HepG2 hepatocellular carcinoma cell nuclei. (b) For four individual cell nuclei, the left, right, top, and bottom boundaries are shown as green, yellow, pink, and red, respectively.

In this thesis, we introduce a new marker-controlled watershed algorithm that incorporates human perception into cell segmentation. This “perceptual watershed” algorithm relies on modeling a very trivial fact that a human uses: each cell should have a left boundary, a right boundary, a top boundary, and a bottom boundary and these boundaries should be in the correct position with respect to each other. Fig. 1.2 illustrates these boundaries for four individual cell nuclei with the left, right, top, and bottom boundaries being shown as green, yellow, pink, and red, respectively. In the figure, one can observe that the bottom (red) boundary of Cell 1 is not identified due to an uneven lighting condition. Similarly, the right (yellow) boundary of Cell 2 is not present in the image due to its partial overlapping with Cell 3. However, it is possible to identify these two cells by

using only their present boundaries and the spatial relations of these boundaries. Moreover, here it is obvious that the green boundary of Cell 3 cannot belong to Cell 2, but it reveals the fact that Cell 3 is overlapping with Cell 2. On the other hand, the boundaries of Cell 3 and Cell 4 form close shapes, which make them easy to separate from the background. In this thesis, such observations are our main motivations behind implementing the proposed perceptual watershed algorithm.

1.2 Contribution

In the proposed algorithm, our contributions are three-fold: First, we represent cell boundaries (left, right, top, and bottom) by defining four different types of primitives and represent their spatial relations by constructing an attributed relational graph on these primitives. Second, we reduce the marker identification problem to the problem of locating predefined structural patterns on the constructed graph. Third, we make use of the boundary primitives to guide the flooding process in the watershed algorithm. The proposed algorithm mainly differs from the previous cell segmentation algorithms in the following aspect. Instead of directly working on image pixels, our algorithm works on high-level boundary primitives that better correlate with the image semantics. The use of the boundary primitives help better separate confluent cells. Moreover, this use is expected to be less vulnerable to noise and variations that are typically observed at the pixel level. Working on a total of 2661 cells in two datasets, our experiments demonstrate that the proposed perceptual watershed algorithm, which uses the boundary primitive definition, improves the segmentation of fluorescence microscopy images by locating better markers and obtaining better cell boundaries for both less and more confluent cells, compared to its counterparts.

1.3 Outline

The structure of this thesis is as follows. In Chapter 2, we introduce a brief description of the problem domain including the molecular biological background of this study and specific characteristics of fluorescence microscopy images. We end up the chapter by introducing high content screening with its sub-steps and summarize existing approaches from the literature for cell/nucleus segmentation. In Chapter 3, we give the details of our proposed cell segmentation method including the primitive definition, graph construction, cell localization (marker identification), and region growing steps. In Chapter 4, we present our experimental results with the explanation of datasets, test environment, comparison methods, and parameter analysis. In Chapter 5, we finalize the thesis with a conclusion and discuss the future work.

Chapter 2

Background

2.1 Domain Description

In this section, we introduce a brief description of fluorescence microscopy and the specific characteristics of fluorescence microscopy images. Moreover, we discuss the difficulties in fluorescence microscopy imaging, which have a negative effect on the performance of the segmentation process. The section is ended up with some information about the cell lines used throughout this study and basic preparation details of the specimens before taking the images.

2.1.1 Fluorescence Microscopy Images

Fluorescence microscopy has been a very functional technology in both biological and medical areas. The name “fluorescence microscope” origins from its working principle, in which fluorescence is made use of to generate an image. The specimen that is wanted to be studied is fluoresced so that it can emit the light with a specific wavelength while sorting out the others. Here, the key is usage of filters which only allow the desired wavelength to pass and block the rays of light with undesired wavelengths [12]. Since the specimen is radiated by fluorescence, cellular regions are observed in the microscope to shine out on a dark

background with a high contrast. This high contrast between the foreground and the background arises as sharp intensity changes on the taken images, therefore it becomes possible to identify the cellular regions clearly. In Fig. 2.1, several example fluorescence microscopy images are given, where specific characteristics of the images can be observed clearly.

Separation of cellular regions from the background is relatively easy due to aforementioned characteristics of fluorescence microscopy images. On the other hand, a critical factor for the performance of the segmentation task is the quality of the images. Noise and uneven illumination turn up to be the biggest drawbacks during the segmentation process. These drawbacks mainly arise due to technical issues related with the hardware (a microscope and a camera) and/or the specialist. The focus level of the objective, the effects of the filters, the illumination of the image as well as the setup of the hardware and the experiment play a critical role for obtaining images of high quality. The adjustments should be set in a careful manner, which is directly related with the expertise of the specialist. Moreover, if the image taking process is handled manually, the handiness of the expert who captures the images is also very important. All these factors may result to good-quality, clear, and well-focused images or bad-quality, noisy, and shaded images, which make a huge difference for an accurate segmentation process and the further analysis. An example of uneven illumination is shown in Fig. 2.1(a) and Fig. 2.1(c). A robust segmentation algorithm should be capable of handling such problems or minimize their effects.

The structure of cells, on the other hand, is another factor that may affect the quality of images. Similar to high contrast between cells and the background, the subcellular objects in the cytoplasm and some artifacts related to the specimen may also have a textural difference inside the cells. Although they may be useful as an indicator for identifying cells, they may also mislead the algorithm, which results in false detection of a bright pixel out of a cellular region or oversegmentation. Thus, these artifacts are considered as noise and should be avoided to increase the accuracy of the algorithm. An example noisy image is presented in Fig. 2.1(d).

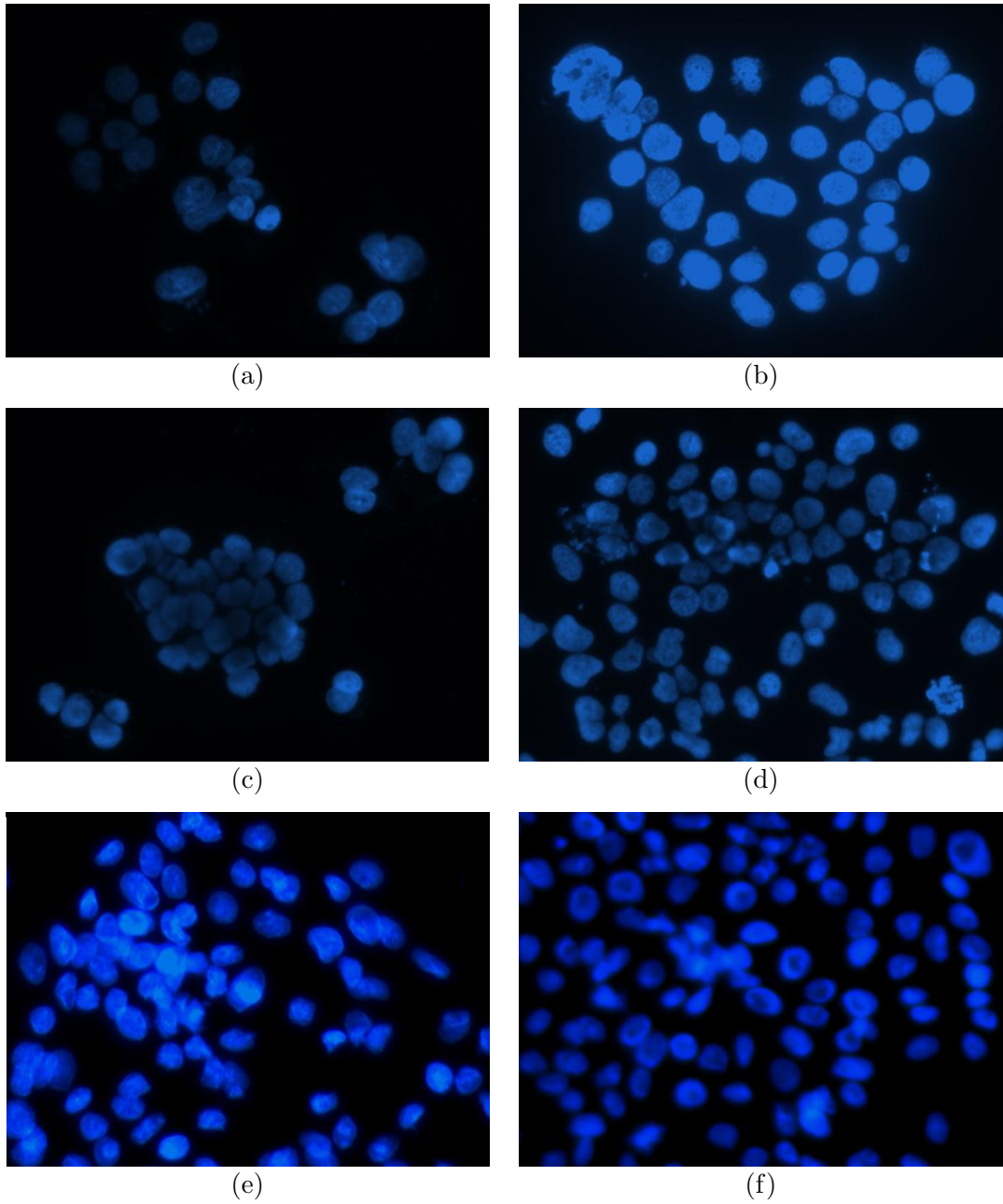


Figure 2.1: Several example fluorescence microscope image segments: (a), (c), (e) are taken from the HepG2 dataset; (b), (d), (f) are taken from the Huh7 dataset. The images evidently share some features such as color, bright foreground and dark background, but also show some differences in texture and illumination due to different cell lines and technical problems emerged during the image acquisition.

Another roadblock in automated cell segmentation via fluorescence microscopy lies on the nature of cell behavior. Although, for example the treated cells in drug screening tend to grow in monolayers, cells naturally grow in aggregates. As they continue to grow and divide, the confluency of the cells increase simultaneously, resulting in a more dense image. Therefore, segmenting overlapping cells turns to be a harder problem compared to segmenting monolayer/isolated cells (see Fig. 1.1).

2.1.2 Cell Lines

In this study, we used two set of images, taken from human hepatocellular carcinoma (HCC) cell lines (Huh7 and HepG2). Sample preparation and image acquisition took place at the Molecular Biology and Genetics Department of Bilkent University. The cells were cultured routinely at 37°C under 5% CO₂ in a standard medium (DMEM) supplemented with 10% FCS. To visualize the selenium deficiency or drug dependent morphological changes in HCC cells, cells were seeded on autoclave-sterilized coverslips in 6-well plates and cultured overnight in the standard medium. Next day, for the induction of cell death through selenium deficiency dependent oxidative stress, cells were exposed to a selenium-deficient (HAM's medium with 0.01% FCS) or a selenium-supplemented (HAM's medium with 0.01% FCS and 0.1 µM sodium selenite (Sigma)) medium. Cells were maintained in these media for up to 10 days by refeeding with fresh media every 2 days. For the visualization of cell death through drug-induced cytotoxicity, cells seeded on 6-well plates were supplemented with fresh media after overnight culture and were treated with two well-known cell death inducing drugs Adriamycin (1 µg/ml) and Camptothecin (5 µM) for 48h.

During the experiments, cell morphology changes were observed under an inverted microscope. To determine nuclear condensation by Hoechst 33258 (Sigma) staining, coverslips were washed with ice-cold PBS twice and fixed in 70% (v/v) cold ethanol for 10 min. After incubation with 1µg/ml Hoechst 33258 for 5 min in the dark, coverslips were destained with ddH₂O for 10 minutes and mounted on glass microscopic slides in 50% glycerol, to be examined under a Zeiss Axioskop

fluorescent microscope.

2.2 High Content Screening

High content screening fuses the efficiency of high-throughput techniques with fluorescence microscopy imaging to process and analyze large datasets in a considerable amount of time. The produced quantitative output accelerates the decision-making process in molecular cellular biology research, particularly in drug discovery. Development of efficient and effective algorithms and latest advances in hardware are the key points behind the high-throughput capability of high content screening tools. On the other hand, the whole system's performance is directly dependent on the effectiveness of sub-systems, each of which has its own importance for qualified analysis.

High content screening typically consists of independent sub-steps, all of which aim to solve specific problems [13]. These sub-steps include acquisition of the images, image preprocessing, cell/nucleus segmentation, cell tracking and registration, feature extraction, data modeling and storage, statistical analysis, and visualization. Figure 2.2 summarizes the flow of the process in a typical high content screening tool.

The image acquisition step provides fluorescence microscopy images that will be processed throughout the system to extract quantitative data. The quality of the images is very critical for better segmentation and consistent feature extraction (refer to Sec. 2.1.1 for more information). Unfortunately no imaging system is perfect, thus the images probably would suffer from noise and/or uneven shading. To reduce the negative effects of uneven illumination, image preprocessing techniques are applied to the images, such as contrast enhancement and noise removal. Next, cell segmentation takes place, which is the core step in image analysis, since the performance of next sub-steps are very dependent on the accurate segmentation of cells. The aim of cell segmentation process is to identify

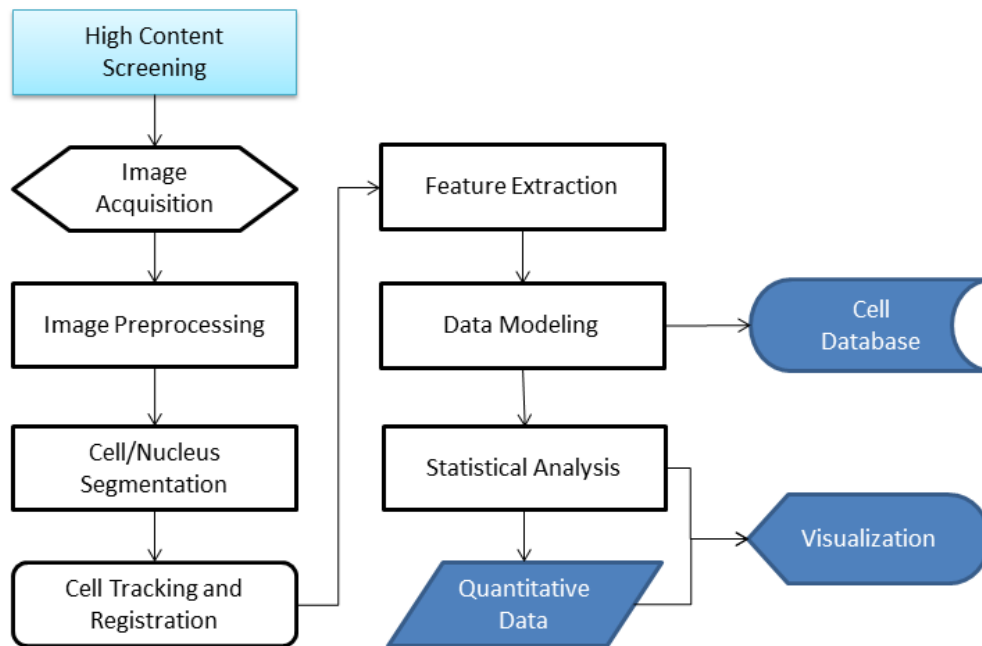


Figure 2.2: High content screening pipeline, which consists of image acquisition, image preprocessing, cell/nucleus segmentation, cell tracking and registration, feature extraction, data modeling and storage, statistical analysis, and visualization.

and separate interest areas, so that cell features can be extracted and quantitative experiments can be conducted on them. Cell segmentation is usually followed by cell tracking and registration in many applications since examining the cell behavior, such as mitosis and the phenotype of a cell would be of high value. Tracking directly works on segmented objects and attempts to associate possible divided cells according to some criteria such as the speed of motion, the shape of the trajectory, and the possibility of being divided from the same parent [14].

After acquiring the cells and tracks between them, the image processing part finishes and the data analysis part starts. For that, the first step is extracting some numeric data to describe cell specific features for further analysis. A variety of features such as area, shape, size, perimeter, intensity, texture, and pattern are widely used to classify different types of cells [15]. On the other hand, track specific features such as the change of the size and shape of the cell during and after mitosis can be useful to identify the track or the behaviour of

a cell [13]. After feature extraction, the high amount of data should be organized and archived properly in a well designed database for statistical feature analysis, cell classification, and data mining, so that valuable information would be extracted. Visualization is also an important step to clearly represent the information gained throughout these processes.

Details of the cell segmentation step and state-of-the-art algorithms as well as some brief information about image preprocessing will be given in the following subsections, while other system steps are not in the scope of this thesis. More information about high content screening is briefly given in [13, 16, 14].

2.2.1 Image Preprocessing

Image preprocessing refers to a series of techniques to improve the quality of raw images before further processing such as segmentation and feature extraction [13]. There exist various preprocessing techniques [17, 18], however the most useful ones for fluorescence microscopy images are noise removal and contrast enhancement.

High amount of intensity variations and nonuniform shades caused by uneven illumination are not desired in images, since they directly affect the performance of image analysis. Besides, the noise, which adds spurious and extraneous information to the images, is also an important problem especially for fluorescence microscopy images (see Sec. 2.1.1). Therefore, the effects of uneven illumination and noise in the images should be reduced before continuing through further processes. Fig. 2.3 illustrates the positive effects of enhancement on a sample fluorescence microscopy image. After processing the image in Fig. 2.3(a), it is observable that the amount of uneven shades reduces in Fig. 2.3(b), while the edges of the inner cells become more clear.

Contrast enhancement, also called shade correction, is the process of correcting illumination artifacts and reducing intensity variations in the images. An

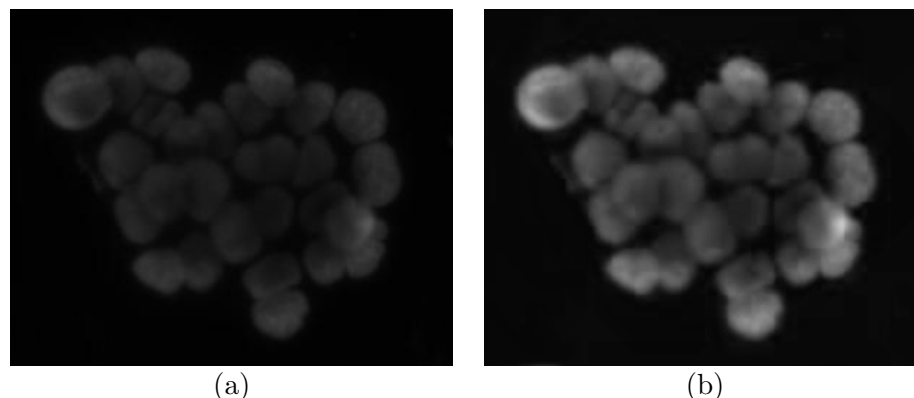


Figure 2.3: Results of applying preprocessing methods to the image: (a) an example fluorescence image with high intensity variations and nonuniform shades due to uneven illumination, (b) the resulting image after preprocessing.

iterative method for this purpose is based on B-spline estimation of the background which improves the quality of the image in each iteration by subtracting the estimated B-splines from the original image [19, 20]. Histogram equalization is another method for contrast adjustment, which distributes the intensities more evenly, therefore reduces the amount of intensity variations in the images [21, 22]. Besides, it is also possible to apply filters beforehand for enhancing the images [23, 24, 6].

Noise removal is the process of eliminating undesired stains, spots, and artifacts from the images, which emerge due to specimen or instruments used while acquiring the images. Noise removal techniques are usually based on convolution, in which a filter or a kernel is convolved on the image removing the noise and smoothing the objects in local segments. Popular techniques used for this purpose include Gaussian filtering [11, 5], median filtering [25, 26, 20], morphological opening [27], and windows slicing [10].

2.2.2 Cell/Nucleus Segmentation

The first step in cellular image analysis is automatic detection and identification of cells/nuclei. For this aim, cells have to be separated from the background by specific image processing techniques. It is in the heart of the cellular image

analysis, regardless of the application area. The accuracy of segmentation process directly affects the results of the study, since the subsequent steps of the analysis are highly dependent on the segmented cells. Therefore, cell segmentation is one of the most intensely studied problems in cellular image analysis. While a robust and general method for cell segmentation is desired in the area, it is difficult to model a generic strategy with the ability to work for all types of cellular images. Therefore, various methods have developed using the information extracted from different characteristics of the images.

The high contrast between fluoresced cells and the dark background leads to sharp intensity changes in the images. These properties play a primary role in the design of cell segmentation methods for fluorescence microscopy. On the other hand, cell segmentation is an ill-posed problem, therefore the frame of the problem should be delineated clearly and the algorithm should make use of the specific characteristics of the images together with the domain knowledge. Otherwise, cells may not be segmented properly or even may not be detected. The segmented regions labeled as cells may correspond to more than one actual cell, which is called undersegmentation, or an actual cell may further be segmented into subregions, which is called oversegmentation. In Fig. 2.4 true, oversegmented, and undersegmented results are illustrated for better understanding.

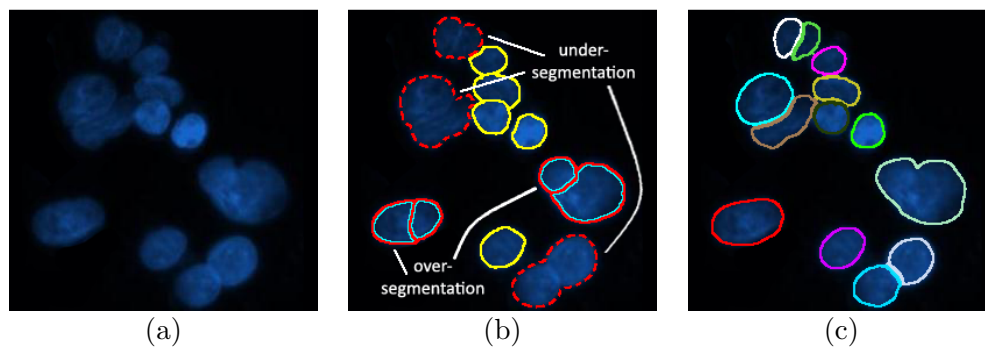


Figure 2.4: Illustrations of true, oversegmented, and undersegmented cells: (a) a sample fluorescence microscopy image, (b) segmentation results, where oversegmented cells are annotated with red-cyan lines, undersegmented cells are annotated with red dotted lines, and true cells are annotated with yellow lines, (c) segmentation delineated by an expert, where each color represents a cell.

Well known techniques that focus on the segmentation problem in fluorescence microscopy images include thresholding, active contours, and watershed-based methods. These methods are usually applied on preprocessed images to increase the quality of segmentation and followed by a post-process to refine the segmentation results. The following sections will briefly explain the methods and the-state-of-the-art algorithms in the literature of cellular image analysis, with their weak and strong points. On the other hand, this thesis commonly focuses on the methods specifically designed for cell/nucleus segmentation. A more extensive study on image segmentation [28] and a survey on segmentation algorithms developed for the medical field [29] can give a deeper understanding to the reader.

2.2.2.1 Thresholding

Thresholding is the simplest method to differentiate foreground objects and the background. The simple idea behind the method is classifying each grayscale pixel according to a predefined threshold. If the intensity of the pixel is above the threshold, it is classified as a foreground pixel, otherwise it is classified as background. Let I be the intensity image and t be the threshold value. The binary image B obtained via thresholding is defined as:

$$B(i, j) = \begin{cases} 1 & \text{if } I(i, j) > t \\ 0 & \text{otherwise.} \end{cases} \quad (2.1)$$

The key point here is the selection of the right threshold. For a better differentiation, a threshold should be preferred which maximizes the inter-class variance and minimizes the intra-class variance between the foreground objects and the background [13]. The threshold may be computed globally, in which the entire image is used [1] or, adaptively, where the threshold is computed using the local information gained from the sub-regions of the image [30]. A detailed study on image thresholding methods is presented in [31].

In fluorescence microscopy images, cellular regions are bright objects that

shine out on a relatively darker background. Therefore, it is easy to differentiate the cellular regions and the foreground via thresholding. If an image contains mostly monolayer isolated cells, finding connected components on the binary map is usually sufficient to locate the cells. However, when the image contains confluent cells as well, thresholding usually leads to undersegmentation (Fig. 2.4(b)). To overcome this problem, more complex techniques are necessary to separate the confluent cells and these techniques can use the binary map in the subsequent system steps (for instance, the binary map is used to calculate a distance transform for the watershed algorithm, which will be shortly covered in 2.2.2.3). Therefore, for fluorescence microscopy images, the thresholding result is generally used as a binary map or a mask for more sophisticated segmentation algorithms, rather than identifying individual cells [32, 3, 33, 34].

When the image is uniformly illuminated, it is well known that global thresholding [35, 36] is a simple and good way to obtain a binary map that differentiates cellular regions and the image background. On the other hand, applying global thresholding on images caused with uneven illumination usually does not lead to desired results. To overcome this, local (adaptive) thresholding is used, so that pixels are classified according to locally calculated thresholds so that the other parts of the image are not affected [4, 37, 38]. In Fig. 2.5, the results of applying global and local thresholding to a sample image is illustrated. The figure reveals that local thresholding is always one step forward from global thresholding when the image is not uniformly illuminated.

2.2.2.2 Active Contour Based Methods

Active contours, also called snakes and deformable models [39], have been widely used to delineate the object outline since they first introduced in [40]. An active contour is a spline, localized to an object boundary by minimizing its energy function. This function is defined on internal forces that control the smoothness of the boundary as well as external forces that pull the boundary towards the object's gradient. The dynamic behaviors of active contours give them the ability to find the object boundaries even in noisy environment. It is also possible to

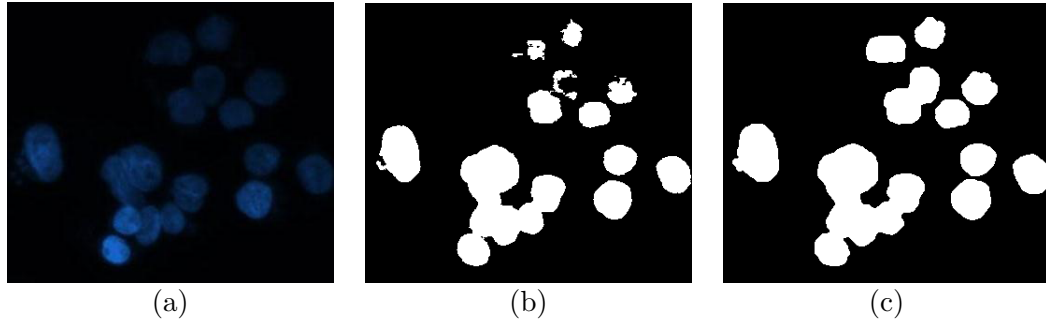


Figure 2.5: The results of applying global and local thresholding to an example image: (a) a sample fluorescence microscopy image, (b) the segmentation result of global thresholding, (c) the segmentation result of local thresholding. Due to uneven illumination, global thresholding classifies relatively darker cellular pixels as background.

increase the power of snakes by introducing different energy functions [41]. On the other hand, it is very critical for snakes to start with a good initial spline for obtaining good final results. Most of the time, the initial spline points near the object boundary are provided by a user interface or approximate object coordinates are given to the snake by an initial segmentation map. How snakes converge into object boundaries is illustrated on a sample fluorescence image in Fig. 2.6.

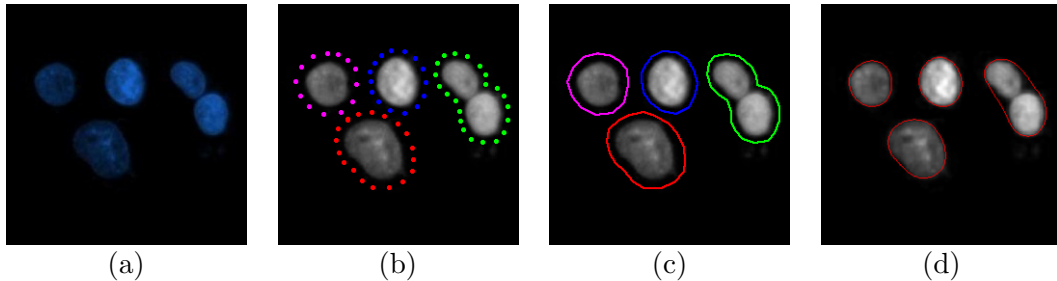


Figure 2.6: Illustration of how snakes work: (a) a sample fluorescence image, (b) initial points provided around object boundaries, (c) initial boundaries, (d) final segmentation results. When initial points are given properly, the spline delineates the object boundaries, but with some limitations: (i) the spline may not localize concave curves accurately, (ii) touching cells may not be separated, which yields undersegmentation.

Active contours have been fully automated since gradient vector flow was introduced [42] to guide the spline as an external force. This allows the active

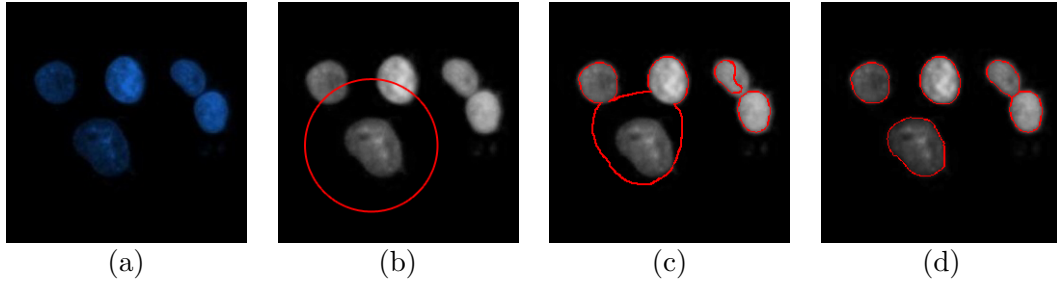


Figure 2.7: Illustration of active contours without edges: (a) a sample fluorescence image, (b) the randomly assigned first spline, (c) splines after a few hundred iterations, (d) final segmentation results. Active contours without edges can converge into boundaries regardless of the initial points. Contrary to the snakes as illustrated in Fig. 2.6, it better manages to separate touching cells.

contour to find its way through the interest points. Furthermore, the effectiveness of active contours upgraded to a higher level when their dependencies on the gradient were broken by active contours without edges [43]. It is a single level set algorithm that divides the image into two regions. The spline is evolved by the average intensity that is computed from the segmented regions that remain inside and outside of it. The method gives quite accurate results even if foreground objects show similar textural properties. Moreover, neither the initial curve needs to be around the object boundaries, nor it should be initialized externally. The segmentation process handled by active contours without edges is illustrated in Fig. 2.7.

Deformable shapes of cells and intensity variations in the images are very suited to be modeled by active contours, thus several active contour based models were adapted to solve the cell segmentation problem in fluorescence microscopy images. Some of them follow an updated way of the traditional active contours which rely on the gradient [44, 45, 46] or the intensity [47]. The snake-based methods make use of a binary map obtained via thresholding [45], a modified version of the gradient vector flow [46], or a mixture of them [44] to initialize object boundaries for curve evolution while the region-based methods rely on a level set based active contour algorithm [47]. The common point of these studies is the affinity of the images where the cells have strong gradients which differentiate them from the background and the other objects.

On the other hand, traditional active contour models do not yield good results when images mostly consist of overlapping or touching cells, so that, they tend to merge adjacent boundaries into single objects. In order to overcome this drawback, a multiphase extension of a traditional single level set method [48], that defines $\log n$ level set functions for n phases has been presented in [20]. Several studies carried this idea further by defining level set functions for every cell on the initial segmentation map and defining coupling constraints to prevent adjacent cells from merging [49, 50, 51]. The reason behind using different level sets for every cell is that the average intensities inside cells do not need to be equal. Thus, objects with different average intensities are segmented more accurately. Moreover, by assigning penalties to contour overlaps, merging of adjacent cells is avoided. Different from its ancestor, multiphase level set methods need initial contours, which may be obtained via thresholding [20] or an automatic initialization using a single level set function [49].

2.2.2.3 Watershed Based Methods

For more than two decades, watershed algorithms have been one of the primary tools for image segmentation [52, 53, 54]. The method is inspired from *watersheds* of the field of topography, which are geographical boundaries (i.e., ridges) that divide adjacent catchment basins [53]. In mathematical morphology [55], gray-scale intensity images or gradient magnitudes can be interpreted as topographic surfaces, in which bright and dark pixels correspond to hills and hollows, respectively. Each intensity or gradient in the image represents the altitude of that point in the landscape where the peaks are the pixels with high intensities, or high gradients (i.e., the edges of the objects), and vice versa. To better understand the watershed algorithm, imagine that a rain starts over the landscape, filling the minima with water. As the water rises, water in adjacent catchment basins meet and the points where two catchment basins join each other form a watershed which, at the end, corresponds to the segmented object boundaries. In Fig. 2.8, the immersion of the landscape and the formation of the watersheds are illustrated.

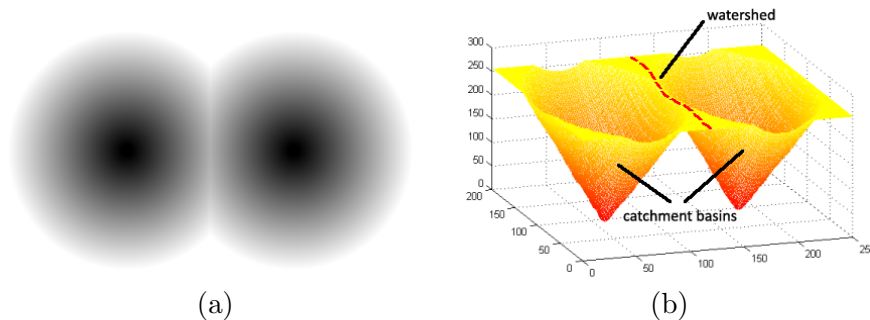


Figure 2.8: Illustration of a watershed in the field of image processing: (a) a synthetically generated gray scale image of two dark blobs, (b) 3D surface plot of the intensities where the colors of the points in the space turn into yellow as the intensity of the pixels increases. Starting from the minima (the darkest red), catchment basins merge onto the watershed line, illustrated with a dotted red line.

Watershed-based algorithms are of great importance to handle the cell segmentation problem. Thresholding and deformable models work well on images that mostly consist of monolayer isolated or touching cells, but their power weakens on the clustered/overlapped nuclei, which typically leads to undersegmented results. Therefore, when nuclei are clustered and/or have fuzzy boundaries, watershed based algorithms come into play. On the other hand, applying the watershed algorithm to only gray-scale intensities or gradient magnitudes always almost leads to oversegmentation due to regional minima in both objects (cells) and the background. The flooding operation starts from the regional minima all over the image, and regardless of their levels, a watershed line is formed as soon as two floods meet, resulting in more segmented objects than expected. An example segmentation result obtained by a watershed algorithm applied to only gradient magnitudes is illustrated in Fig. 2.9. Since regional minima emerge from uneven gradients near boundaries and the centroids of cells, the process ends up with oversegmentation.

A common solution to handle this problem is assigning seed points, instead of letting the flood start from regional minima. This method is called marker-controlled watershed, since the regions start to grow only from previously identified seed points (markers) and the flow of the flood is controlled by a marking function. But this gives another important question: How should one define

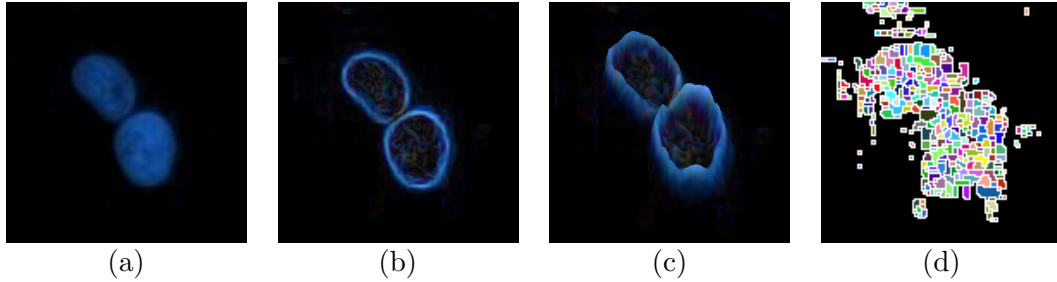


Figure 2.9: An example segmentation result obtained by a watershed algorithm applied to only gradient magnitudes: (a) a sample fluorescence image, (b) the gradient magnitudes, (c) 3D illustration of the gradient image, in which local minima can be observed around boundaries and the centroids of cells, (d) labeled segmented regions after applying watershed to the gradients. Since the flooding starts with all minima at the same time and catchment basins join as soon as they meet, the image is highly oversegmented.

markers so that each of them matches with an existing object? The answer to this question hides under the problem itself. Since oversegmentation occurs due to high number of spurious minima, it would be reasonable to eliminate them. For that, h -minima transform is a widely used method that suppresses undesired minima on the image [7, 56, 11]. On the other hand, applying h -minima transform to reduce the number of false minima does not help solve the oversegmentation problem when gradients or intensities are severely uneven. An example cell that shows such characteristics is given in Fig. 2.10(a). When h -minima applied to the image (Fig. 2.10(d)) the amount of over partitioned parts is highly reduced (Fig. 2.10(e)) compared to the segmentation results where only regional minima are used (Fig. 2.10(c)), but the oversegmentation problem still exists.

Another approach for marker definition is to make use of *a priori* shape information captured via initially segmenting the image [32, 3]. This initial segmentation result, usually referred as a binary map or mask, would be used for marker identification [33, 34]. It is also used for marking function definition [7, 57] to cover all cellular regions so that while the markers are growing the intrusion of floods to other cells and the background are avoided.

One of the widely used methods to integrate the shape information into a marker controlled watershed algorithm is inner distance transform [25, 8]. In

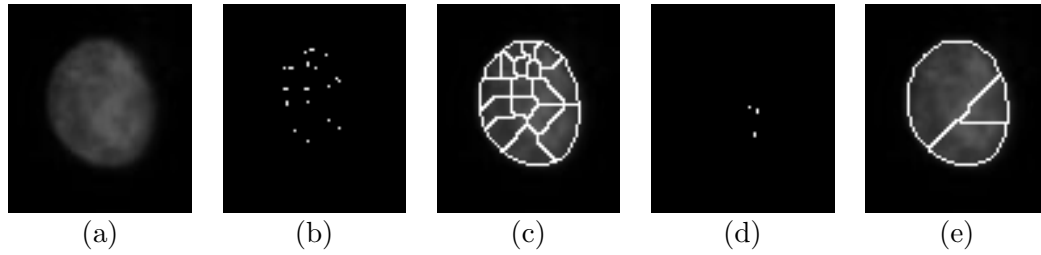


Figure 2.10: An example to show that regional minima and suppressed minima cannot handle oversegmentation problem when intensities are uneven: (a) gray-scale intensity image, (b) minima of the intensity image, (c) segmentation results using regional minima as markers, (d) suppressed regional minima via h -minima transform, (e) segmentation results using suppressed minima as markers. Although h -minima transform is used, the oversegmentation problem may still exist. Note that, for better illustration, a mask was used to eliminate noise in the background.

this transform, for each pixel in the foreground, a value is assigned which is the distance to the nearest zero pixel of the mask, as illustrated in Fig. 2.11. Inner distance transform ensures that the pixels around the centroids get the furthest distances to the background. Reversing the inner distance map (Fig. 2.11(d)), the furthest distances turn up to be the regional minima (or regional maxima for the inner distance transform) in the map, which can be used as markers for the watershed algorithm [58, 7]. Besides, to better deal with the problem, it is also possible to combine shape information with gradient/intensity information [58, 11, 21].

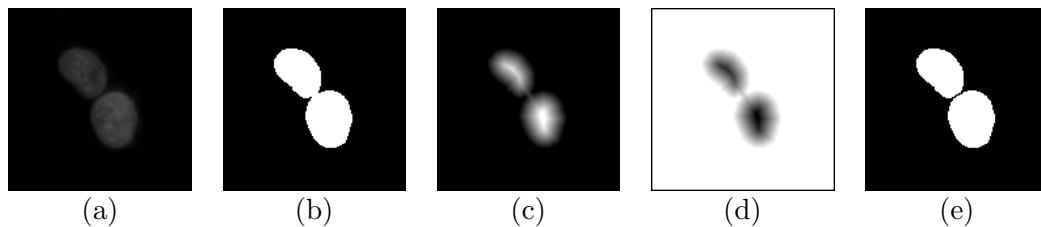


Figure 2.11: Integrating shape information in a marker-controlled watershed via distance maps: (a) the gray scale intensity map of the image in Fig. 2.9, (b) binary map obtained via Otsu's thresholding method [1], (c) the inner distance transform map where the pixels around cell centroids have the furthest distance to the background, (d) reverse of the inner distance transform map where the regional minima corresponds to the markers, (e) the segmentation result after applying marker-controlled watershed algorithm to the binary map.

Another way of using cellular shapes for marker definition is to apply morphological operations to the mask of an image. One of the classical approach for that is using ultimate eroding points as seed points [17]. But since it is inadequate to oversegmentation, iterative approaches have been proposed in which morphological erosion is applied to the binary map of the image by a series of cell-like structural elements to identify markers, while preserving their shapes [9, 59].

Marker-controlled watersheds give accurate results if there is one-to-one mapping between the markers and the actual cells. Otherwise, they may result in under- or over-segmentation. To address this problem, many studies post-process the segmented cells that are obtained by their watershed algorithms. This post-processing relies on extracting features from the segmented cells and the boundaries of the adjacent ones and using these features to validate, merge, or split the cells. For that, they use score-based [26, 25, 6], rule-based [10, 60, 33], and iterative [38, 61, 62] techniques.

2.2.2.4 Other Methods

Besides thresholding, deformable models and the watersheds, shape-based and graph-based methods can be used for segmentation. The shape-based methods use the fact that cells typically have round and convex shapes. For that, they locate circles/ellipses on the binary map of the image to find the initial cell boundaries and refine them afterwards [63, 27]. Alternatively, they find concave points on the binary map and split the map into multiple cells from these points [64, 65]. In cell segmentation studies, graphs are commonly used to merge the oversegmented cells by constructing a graph over the adjacent ones [25, 66] as well as to refine initial segmentation results through the graph cut algorithms [67, 68, 69]. On the other hand, our segmentation method is different than these previous studies in the sense that it uses graphs to represent the spatial relations of the high-level boundary primitives and to define the markers of a watershed algorithm. Note that besides segmentation, it is also possible to use graphs for cell tracking applications [70, 71, 72], which are beyond the scope this thesis.

Chapter 3

Methodology

The proposed algorithm relies on modeling cell nucleus boundaries for segmentation. In this model, we approximately represent the boundaries by defining high-level primitives and use them in a marker-controlled watershed algorithm. This watershed algorithm employs the boundary primitives in its two main steps: marker identification and region growing. The marker identification step is based on using the spatial relations among the primitives. For that, we first construct a graph on the primitives according to their types and adjacency. Then, we use an iterative search algorithm that locates predefined structural patterns on the constructed graph and identify the located structures as markers provided that they satisfy the shape constraints. The region growing employs the primitives in its flooding process. Particularly, it decides in which direction it grows and at which point it stops based on the primitive locations. An overview of the proposed method is given in Fig 3.1. The details of these steps are explained in the next subsections.

3.1 Primitive Definition

In the proposed method, we define four primitive types that correspond to left, right, top, and bottom cell nucleus boundaries. These boundary primitives are

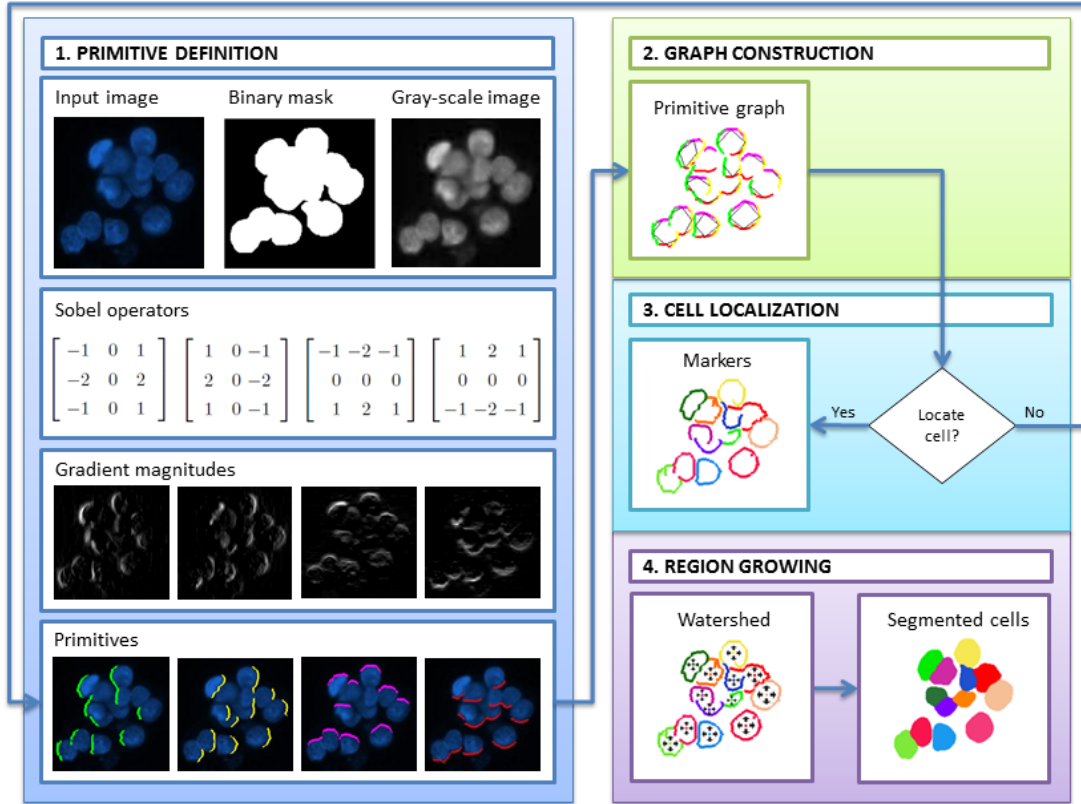


Figure 3.1: Overview of the proposed algorithm.

derived from the gradient magnitudes of the blue band I_b of an image. To this end, we convolve the blue band I_b with each of the following Sobel operators, which are defined in four different orientations, and obtain four maps of the responses. Then, we process each of these responses, as explained below and illustrated in Fig. 3.2, to define the corresponding primitives.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Let R_{left} be the response map obtained by applying the Sobel operator of left orientation to the blue band image I_b . We first threshold R_{left} to obtain a binary left boundary map B_{left} . Here we use local threshold levels instead of using a global one since illuminations and gradients are commonly uneven throughout our

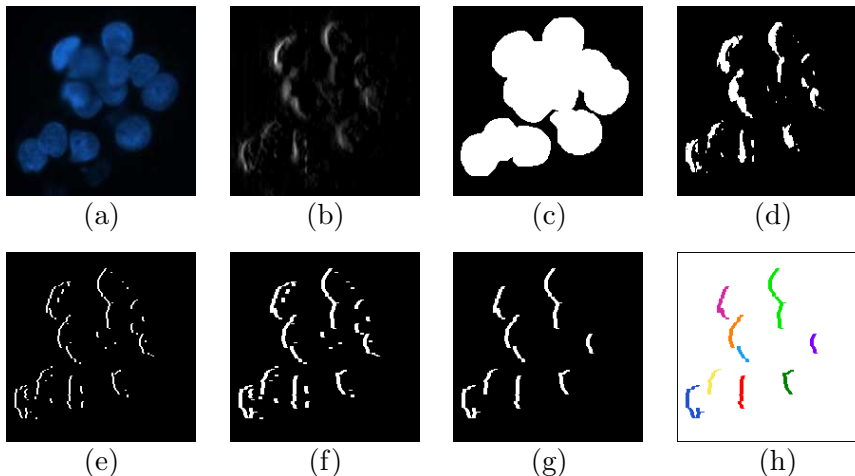


Figure 3.2: Illustration of defining left boundary primitives: (a) original subimage, (b) response map R_{left} obtained by applying the Sobel operator of left orientation, (c) mask that is to be used for determining local Sobel threshold levels, (d) binary image B_{left} after thresholding, (e) boundaries obtained after taking the leftmost pixels, (f) boundary map P_{left} obtained after taking the d -leftmost pixels, (g) P_{left} after eliminating its smaller connected components, (h) left boundary primitives each of which is shown with a different color.

images. For that, we employ a mask that roughly segments cellular regions from the background. For each connected component $C^{(k)}$ of this mask, we calculate a local threshold level $T_{\text{left}}^{(k)}$ on the gradients of its pixels by using Otsu’s method [1]. Then, pixels of this component are identified as boundary if their responses are greater than the calculated local threshold.

Next, we fill the holes in B_{left} and take its d -leftmost pixels. The map P_{left} of the d -leftmost pixels are defined as:

$$P_{\text{left}}(i, j) = \begin{cases} 1 & \text{if } B_{\text{left}}(i, j) = 1 \text{ and} \\ & \exists x \in \mathbb{Z}^+ \text{ s.t. } x \leq d \text{ and } B_{\text{left}}(i - x, j) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.1)$$

In this definition, the d -leftmost pixels are taken instead of just taking the leftmost pixels. The reason is that, as illustrated in Fig. 3.2(e), the leftmost pixels do not always contain all of the cell boundaries, and thus, there may exist discontinuities between the boundaries of the same cell. On the other hand, by taking the d -leftmost pixels, it is more likely to eliminate the discontinuities, as

shown in Fig. 3.2(f). Finally, we eliminate the connected components of P_{left} whose height is less than a threshold t_{size} and identify the remaining ones as left boundary primitives. For an example subimage, Fig. 3.2(h) shows the identified left primitives with different colors.

Likewise, we define the right boundary primitives P_{right} , top boundary primitives P_{top} , and bottom boundary primitives P_{bottom} . In each of these definitions, Eqn. 3.1 is modified as follows:

$$P_{\text{right}}(i, j) = \begin{cases} 1 & \text{if } B_{\text{right}}(i, j) = 1 \text{ and} \\ & \exists x \in \mathbb{Z}^+ \text{ s.t. } x \geq d \text{ and } B_{\text{right}}(i + x, j) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.2)$$

$$P_{\text{top}}(i, j) = \begin{cases} 1 & \text{if } B_{\text{top}}(i, j) = 1 \text{ and} \\ & \exists x \in \mathbb{Z}^+ \text{ s.t. } x \leq d \text{ and } B_{\text{left}}(i, j - x) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.3)$$

$$P_{\text{bottom}}(i, j) = \begin{cases} 1 & \text{if } B_{\text{bottom}}(i, j) = 1 \text{ and} \\ & \exists x \in \mathbb{Z}^+ \text{ s.t. } x \geq d \text{ and } B_{\text{bottom}}(i, j + x) = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (3.4)$$

Eqn. 3.2, Eqn. 3.3 and Eqn. 3.4 give the d -rightmost, d -topmost, and d -bottommost pixels, respectively. Besides, in the elimination of smaller primitives, the components whose height is less than the threshold t_{size} are eliminated for P_{left} and P_{right} whereas those whose width is less than t_{size} are eliminated for P_{top} and P_{bottom} .

In this step, we use a mask to calculate local threshold levels. This mask roughly identifies the cellular regions but does not provide their exact locations. Here our framework allows using different binarization methods such as adaptive thresholding [37] and active contours without edges [43]. However, since this mask is used just for calculating the local thresholds, we prefer using a relatively

simpler method. In this binarization method, we first suppress local maxima of the blue band image I_b by subtracting its morphologically opened image from itself. This process removes the noise from the image without losing local intensity information. Then, we calculate a global threshold level on the suppressed image using the Otsu’s method [1]. In order to ensure that almost all of the cellular regions are covered by the mask, we decrease the calculated level to its half and threshold the suppressed image. Finally, we eliminate small holes and regions from the mask. The benefits of using a mask is illustrated in Fig. 3.3. Global thresholding of the entire image leads to loss of information, thus one can get more accurate results by considering only pixels of interest. The binary mask also avoids spurious bright pixels inside cellular regions.

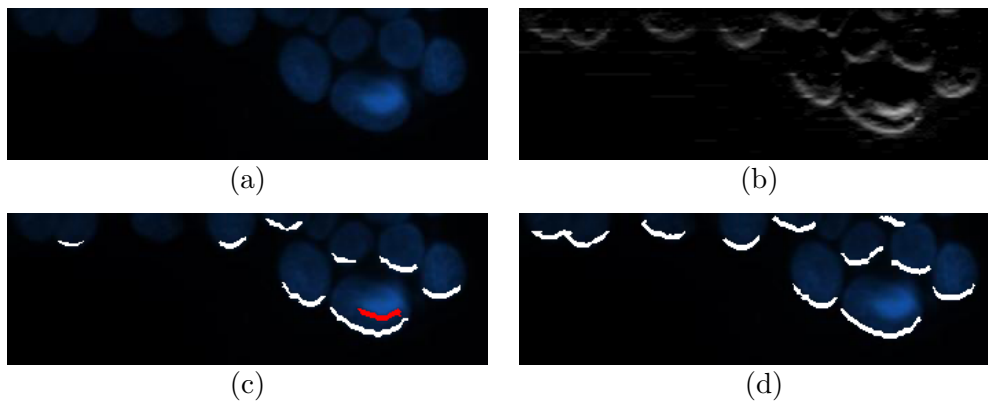


Figure 3.3: Illustration of the benefits of using a mask: (a) original subimage, (b) response map R_{bottom} obtained by applying the Sobel operator of bottom orientation, (c) bottom boundary primitives obtained without a mask (a falsely detected primitive is marked with red), (d) bottom boundary primitives obtained with a mask.

3.2 Marker Identification

Markers are identified by first constructing a graph on the primitives and then applying an iterative algorithm that searches this graph to locate structural patterns conforming to the predefined constraint. In the following subsections, we will explain the graph construction step and the iterative search algorithm.

3.2.1 Graph Construction

Let $G = (V, E)$ be a graph constructed on the primitives $V = \{P_{\text{left}}, P_{\text{right}}, P_{\text{top}}, P_{\text{bottom}}\}$ that are attributed with their primitive types. An edge $e = (u, v) \in E$ is assigned between primitives u and v if they satisfy the following three conditions:

1. The primitives should have overlapping or adjacent pixels.
2. One primitive should be of the vertical (left or right) type and the other should be of the horizontal (top or bottom) type.
3. Each primitive should have a large enough segment that lies in the correct side of the other primitive. For left and right primitives, the width of this segment should be greater than the threshold t_{size} , which is also used to eliminate small components in the previous step. Likewise, for top and bottom primitives, the height of the segment should be greater than t_{size} .

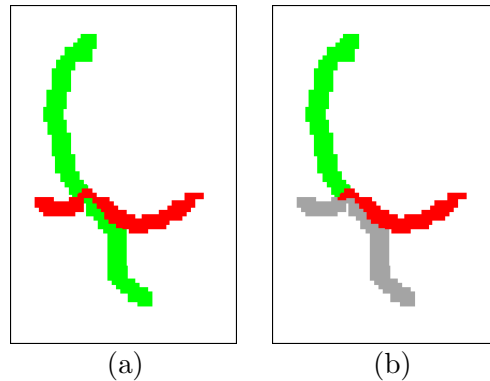


Figure 3.4: Illustration of assignment of an edge between a left and a bottom primitive: (a) primitives and (b) selected segments of the primitives.

Fig. 3.4 illustrates the third condition on an example. In this figure, suppose that we want to decide whether or not to assign an edge between the left primitive u and the bottom primitive v , which are shown in green and red in Fig. 3.4(a), respectively. For that, we first select the segment of each primitive that lies in the correct side of the other. It is obvious that left boundaries of a given cell should be on the upper hand side of its bottom boundaries, and likewise, bottom

boundaries should be on the right hand side of its left boundaries. To model this fact, we select the segment S_u of the primitive u (which corresponds to left boundaries) that is found on the upper hand side of v (which corresponds to bottom boundaries). Similarly, we select the segment S_v of v that is found on the right hand side of u . Fig. 3.4(b) shows the selected segments in green and red, respectively; here unselected parts are shown in gray. At the end, we assign an edge between u and v if both the height of S_u and the width of S_v are greater than the threshold t_{size} .

3.2.2 Iterative Search Algorithm

Each iteration starts with finding boundary primitives, as explained in the primitive definition step given in Sec. 3.1. In that step, the local threshold levels $\mathcal{T} = \langle T_{\text{left}}, T_{\text{right}}, T_{\text{top}}, T_{\text{bottom}} \rangle^1$ are calculated and pixels with the Sobel responses greater than the corresponding thresholds are identified as boundary pixels, which are then to be further processed to obtain the boundary primitives, as also explained in Sec. 3.1.

Our experiments reveal that primitives identified by using the threshold vector \mathcal{T} do not always cover all of the cell boundaries in an image. This is attributed to the fact that illumination and gradients are not even throughout the image (even in the same connected component). For instance, the boundary gradients of cells located closer to the image background are typically higher than those of cells located towards a component center. When the thresholds are decreased to also cover lower boundary gradients, some false primitives can also be found especially inside cells with higher boundary gradients, which causes misleading results. Thus, to consider lower boundary gradients while avoiding false primitives, we apply an iterative algorithm that uses different thresholds in its different iterations. For that, we start with the threshold vector \mathcal{T} and decrease it by its 10 percent at every iteration. Thus, additional primitives, with lower boundary

¹These thresholds are calculated separately for each connected component $C^{(k)}$ of the binary mask. Thus, the notation should be $\mathcal{T}^{(k)} = \langle T_{\text{left}}^{(k)}, T_{\text{right}}^{(k)}, T_{\text{top}}^{(k)}, T_{\text{bottom}}^{(k)} \rangle$. However, for better readability, we will drop (k) from the terms unless its use is necessary.

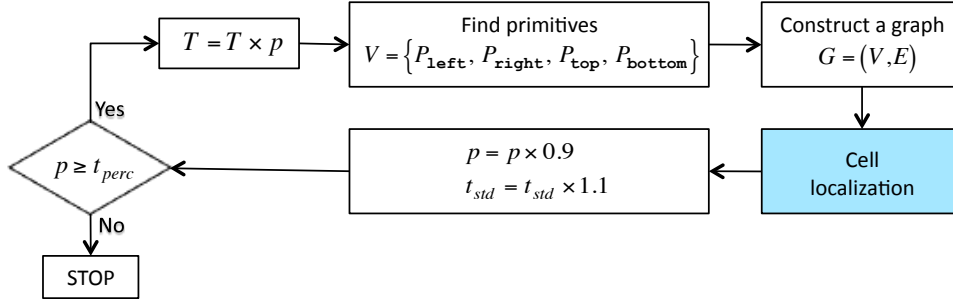


Figure 3.5: Flowchart of the iterative search algorithm.

gradients, can be found at next iterations. This algorithm continues until the decrease percentage p reaches the threshold t_{perc} , which is an external parameter. Here note that p is initially set to 1, which implies that the algorithm always uses the initial threshold vector \mathcal{T} in its first iteration.

At each iteration, a graph is constructed on the identified primitives, as explained in Sec 3.2.1. Afterwards, predefined structural patterns are searched over this graph to locate cells in an image. The cell localization step, which will be explained in the next subsection, locates only cells that satisfy a constraint, leaving the others to next iterations. Since more primitives are found at later iterations, it is possible to find more precise locations of the remaining cells. The flowchart of the search algorithm is given in Fig. 3.5. In this flowchart, t_{std} corresponds to another threshold that defines the constraint in the cell localization step. To find more cells at later iterations, this threshold is also relaxed by its 10 percent at every iteration. The details of the constraint (and the threshold) will also be given in the next subsection.

3.2.3 Cell Localization

Cells are located by searching two structural patterns on the constructed graph. These are 4PRIM and 3PRIM patterns. The 4PRIM pattern consists of four adjacent primitives, each of which is of a different type, and the edges in between these primitives. In other words, this pattern should consist of one left, one right, one top, and one bottom primitive that form a connected subgraph. Some instances

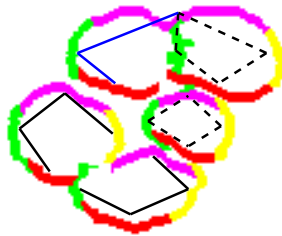


Figure 3.6: Two structural patterns used for cell localization: 4PRIM and 3PRIM patterns. Some instances of these patterns are shown in this figure, indicating the corresponding edges with black and blue, respectively. Moreover, the 4PRIM pattern has two subtypes that correspond to subgraphs forming a loop (dashed black edges) and a chain (solid black edges).

of this pattern are shown in Fig. 3.6, indicating their edges with black. As also observed in this figure, there can be two different subtypes of this pattern. The first subtype (shown with dashed black edges) corresponds to the ideal case where all of the cell boundaries have high gradients. The second subtype (shown with solid black edges) corresponds to a less ideal case where some parts of the boundaries do not have high enough gradients, most probably because of the corresponding cell being confluent with the others. The 3PRIM pattern corresponds to a much less ideal case, in which one of the boundary types cannot be identified at all; this boundary type can be missing due to confluency of cells or some non-ideal lighting conditions. Hence, the 3PRIM pattern should contain three adjacent primitives, each has a different type, and the edges in between these primitives. An instance of this pattern is also shown in Fig. 3.6, indicating its edges with blue.

The cell localization step starts with searching instances of the 4PRIM pattern on the constructed graph $G = (V, E)$, which of course may contain more than one of such instances. Hence, the proposed localization method selects the “best” instance that satisfies the shape constraint, takes the segment of each primitive that lies in the correct sides of the others (as explained in Sec. 3.2.1 and illustrated in Fig. 3.4), updates the primitive vertices V and the edges E of the graph, and continues with the next best instance. The localization process continues until no instance that satisfies the shape constraint remains. After that, this step continues with searching instances of the 3PRIM pattern in a similar way. The flowchart of this localization method is given in Fig. 3.7.

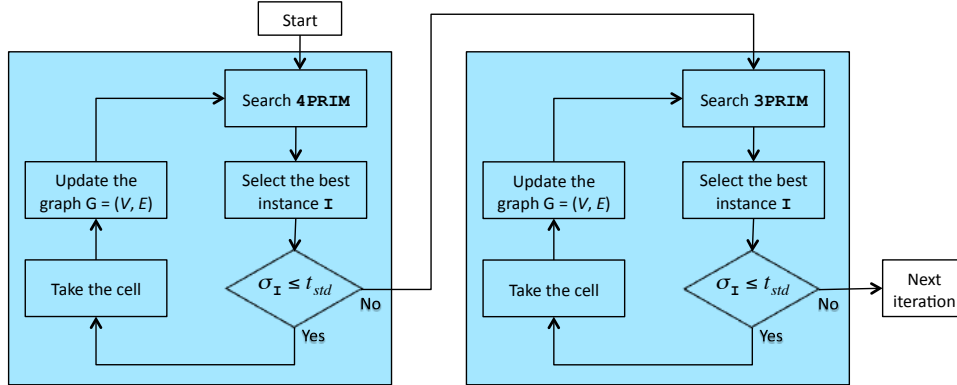


Figure 3.7: Flowchart of the cell localization method.

The shape constraint is defined to process round and more regular shaped cells in first iterations of the search algorithm (Sec. 3.2.2) and deal with more irregular shapes later. Irregular shapes in previous iterations can turn into round shapes in later iterations, in which additional primitives can be found. Moreover, this constraint is relaxed so that it is possible to process more irregular shaped cells in later iterations. In order to quantify the shape of an instance (cell candidate), we use the standard deviation metric. For that, we first identify the outermost pixels o_i of the selected primitive segments (the union of the leftmost, rightmost, topmost, and bottommost pixels of the left, right, top, and bottom primitive segments), calculate the radial distance r_i from every pixel o_i to the centroid C of the outermost pixels, and then calculate the standard deviation σ of the radial distances r_i (Fig. 3.8). This standard deviation is close to zero for round shapes and becomes larger for more irregular ones. The “best” cell candidate is the one that has the smallest standard deviation. Moreover, to impose the shape constraint, we define a threshold t_{std} . If the standard deviation of the best candidate is greater than this threshold, we stop searching the current structural pattern. As mentioned before, this threshold is relaxed by its 10 percent at every iteration of the search algorithm so that more irregular shaped cells can be identified.

After selecting the best instance (cell candidate), we remove its selected primitive segments from the primitive maps and update the graph edges. That is, for the instance shown in Fig. 3.8, the selected segments shown in green, yellow,

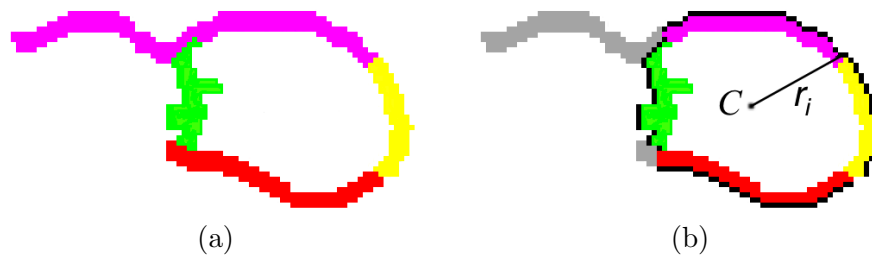


Figure 3.8: Illustration of finding outermost pixels and calculating a radial distance r_i . The primitive segments that lie in the correct sides of the other primitives are identified and outermost pixels are selected. In this figure, unselected segments of the primitives are indicated as gray.

pink, and red and the edges in between them are removed whereas the unselected segments shown in gray are left in the primitive maps. Afterwards, graph edges on the unselected segments are redefined, if necessary.

3.3 Region Growing

The previous step identifies primitives of the located cells. It is straightforward to delineate an individual cell if its primitives form a closed-shape. Nevertheless, most of the time, the identified primitives of a cell do not form a closed-shape due to noise and confluency. In this case, one may think of connecting the end points of the primitives by a line. However, this might give unnatural boundaries especially for 3PRIM instances. Moreover, there may exist some primitives that were incorrectly assigned to a particular cell by the previous step. These primitives should also be corrected in cell delineation. For example, for a subimage given in Fig. 3.9(a), primitives of different cells are shown in different colors in Fig. 3.9(b). Here it can be observed that most of the cells do not have a closed-form. Besides, the top primitive of the red cell (indicated with an arrow) is incorrectly identified; this primitive also contains that of the blue cell next to the red one.

Thus, in order to delineate cells, we use a marker-controlled watershed algorithm. In this algorithm, we take the centroids of the primitives of every cell as markers and grow the markers considering the location of the primitives. We use

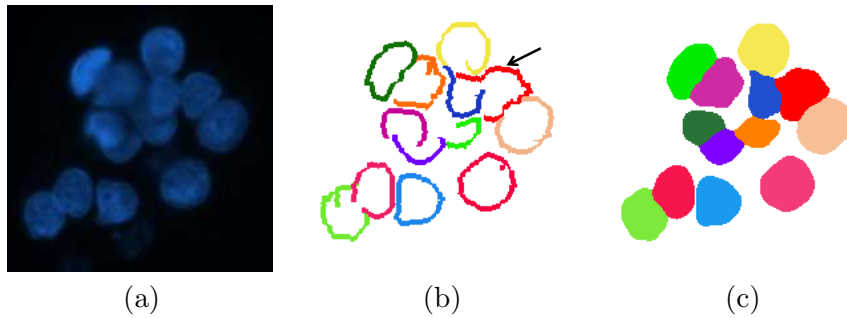


Figure 3.9: (a) Original subimage, (b) primitives identified for different cells, and (c) cells delineated after the watershed algorithm.

the primitive locations in two ends: First, a pixel cannot flood in the direction of the outer boundaries of a primitive that are determined by the primitive type; e.g., a pixel cannot flood to the top outer boundaries of the top primitive. Second, for each cell, pixels reached after the last point that the algorithm floods to a primitive pixel for this cell are excluded from the results. This is effective to stop growing the cells without using an additional mask as well as to obtain better boundaries. In this watershed algorithm, we use the geodesic distance from a pixel to a marker as the flooding criterion. At the end, to obtain smoother boundaries, we apply majority filtering, with a filter radius of W , on the results and fill the holes inside the located cells. For the example subimage, results obtained by this growing step are shown in Fig. 3.9(c).

Chapter 4

Experiments

4.1 Dataset

We conduct our experiments on two datasets of fluorescence microscopy images. These sets include the images of human hepatocellular carcinoma (Huh7 and HepG2) cell lines. The amount of cells growing in aggregates is much higher in the HepG2 cell line. Therefore, the cells in the HepG2 dataset happen to be more confluent than the Huh7 cells. This confluency makes the segmentation problem harder for the HepG2 dataset. To understand the effectiveness of our proposed algorithm on both less confluent and more confluent cells, we test it on both datasets separately and observe the segmentation performance. These datasets contain a total of 2661 cells in 37 images, 16 of which belong to the Huh7 and 21 of which belong to the HepG2 cell lines. The Huh7 and HepG2 datasets include 1378 and 1283 cells, respectively.

Both of the cell lines were cultured and their images were taken at the Molecular Biology and Genetics Department of Bilkent University. These cell lines were cultured routinely at 37 °C under 5% CO₂ in a standard medium (DMEM) supplemented with 10% FCS. For visualization, nuclear Hoechst 33258 (Sigma) staining was used. The images were taken at 768 × 1024 resolution, under a Zeiss Axioscope fluorescent microscope with an AxioCam MRm monochrome camera

using a 20 \times objective lens. We compare automated segmentation results with manually delineated gold standards in which cells were annotated by our biologist collaborators.

4.2 Comparisons

We use two previous algorithms for comparing our proposed method: adaptive h -minima [7] and conditional erosion [9]. Both of the algorithms rely on the shape information to identify markers corresponding to the cells and apply a marker-controlled watershed to initial segmentation maps that they find for separating cells from the background.

Adaptive h -minima [7] introduces a method to identify markers on the initial segmentation of cells obtained via active contours without edges and defines a new marking function to grow these markers in a watershed algorithm. Markers are generated by using an inner distance map that represents the distances of every foreground pixel to the background. The h -minima transform is then applied to the inverse of this inner distance map to suppress undesired minima. The h parameter is selected adaptively on the initial segmentation map and the regional minima obtained after the h -transform are identified as the markers. The marking function is based on the outer distance transform in which the distances from foreground pixels to the nearest marker are calculated. A marker-controlled watershed algorithm is then applied to the combination of the outer distance map and the gray-scale intensity image.

Conditional erosion [9] is based on mathematical morphology, which makes use of a binary image obtained by histogram thresholding. The connected components of the binary image are eroded iteratively by a series of two cell-like structural elements one of which is larger than the other. First, the components are eroded by the larger structuring element until the sizes of the eroded components fall below a predefined threshold. Due to the size of this structuring element and its round shape, the shapes of the components are coarsely preserved.

Next, a second erosion operator with the smaller structuring element is applied on the remaining components to obtain the markers. Likewise, the iterations are stopped just before the sizes of the components get smaller than a second threshold. Considering these components as the markers, a marker-controlled watershed algorithm is applied on the binary image.

4.3 Evaluation

In our experiments, we evaluate segmentation results quantitatively and visually. Here we use two quantitative evaluation methods: cell-based and pixel-based. In the cell-based evaluation, the aim is to assess the accuracy of an algorithm in terms of the number of correctly segmented cells. A cell is said to be correctly segmented if it *one-to-one matches* with an annotated cell in the gold standard. For that, we first match each computed cell to an annotated one and vice versa; a computed cell C is matched to an annotated cell A if at least half of C 's area overlaps with A . After that, the segmented cells that form unique pairs with annotated ones are counted as one-to-one matches, using which cell-based *precision*, *recall*, and *F-score* measures are calculated as follows.

$$\begin{aligned}
 \textit{Precision} &= \frac{\textit{One-to-one matches}}{\textit{Computed cells}} \\
 \textit{Recall} &= \frac{\textit{One-to-one matches}}{\textit{Annotated cells}} \\
 \textit{F-score} &= 2 \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}}
 \end{aligned}$$

Additionally, we consider *oversegmentations*, *undersegmentations*, *misses*, and *false detections* in this evaluation. An annotated cell is oversegmented if two or more computed cells match to this annotated cell, and for the contrary, annotated cells are undersegmented if they match to the same computed cell. A computed cell is a false detection if it does not match to any annotated cell, and similarly, an annotated cell is a miss if it does not match to any computed cell.

In the pixel-based evaluation, the aim is to assess the accuracy of an algorithm in terms of the areas of correctly segmented cells. Here we use one-to-one matches found in the cell-based evaluation and consider the overlapping pixels of the computed-annotated cell pairs of these matches as true positive. Then, pixel-based *precision*, *recall*, and *F-score* measures are calculated. The precision and recall are computed differently from the previous evaluation method. The new definitions are as follows:

$$Precision = \frac{\textit{True positive pixels}}{\textit{Computed pixels}}$$

$$Recall = \frac{\textit{True positive pixels}}{\textit{Annotated pixels}}$$

4.4 Parameter Selection

The proposed algorithm has four external model parameters: 1) the primitive length threshold t_{size} in primitive definition and graph construction, 2) the percentage threshold t_{perc} in the iterative search algorithm, 3) the standard deviation threshold t_{std} in cell localization, and 4) the radius W of the structuring element of the majority filter in region growing. In the experiments, we selected them as $t_{size} = 15$ pixels, $t_{perc} = 0.3$, $t_{std} = 4.0$, and $W = 5$. We will discuss the selection of these parameters and their effects to segmentation performance in the next section. Additionally, there is an internal parameter d that is used to define boundary primitives by taking d -outermost pixels of the binary maps of the Sobel responses. Smaller values of d are not enough to put the boundaries of the same cell under the same primitive whereas larger values cause to connect the boundaries of different cells. Thus, we internally set $d = 3$, which gives good primitives in our experiments.

4.5 Results

We present visual results on example subimages obtained by the proposed perceptual watershed algorithm as well as the comparison methods in Fig. 4.1. Note that these subimages have different sizes, which are scaled for better visualization. Also note that the size of the original images from which they are cropped is the same and we run the algorithms on the original-sized images. Fig. 4.1 demonstrates that all algorithms can segment monolayer isolated cells accurately (the first row of the figure). However, compared to the others, the proposed algorithm gives better results to segment touching and more confluent cells. For these cells, the other methods commonly lead to more undersegmentations, as also observed in quantitative results. For better representing the capabilities of our proposed method, we present visual segmentation results for several other subimages in Figs. 4.5 - Fig. 4.8 at the end of this chapter. These results also reveal that the proposed method can handle the segmentation problem in different types of cells.

We report quantitative results in terms of computed-annotated cell matches for the Huh7 and HepG2 datasets in Tables 4.1 and 4.2, respectively. Additionally, for these datasets, we provide the cell-based precision, recall, and F-score measures in Tables 4.3 and 4.4, as well as the pixel-based precision, recall, and F-score measures in Tables 4.5 and 4.6. These tables show that the proposed perceptual watershed algorithm improves segmentation performance for both of the datasets. The improvement becomes higher for the HepG2 dataset, in which cells are more confluent. This is indeed consistent with the visual results of the algorithms. The improved performance is attributed to the following properties of our algorithm: First, both conditional erosion and adaptive h -minima heavily rely on an initial segmentation map to find their markers. If this map is not accurately found, which is indeed mostly the case, or if there are no background pixels inside a cell cluster, they are inadequate to separate cells in relatively bigger cell clusters. This can be observed in the results of these algorithms given in the second and third rows of Fig. 4.1. Here the problem is more obvious in the results of conditional erosion, which uses global thresholding to obtain its map. On the other hand, our algorithm uses an initial segmentation map to find local

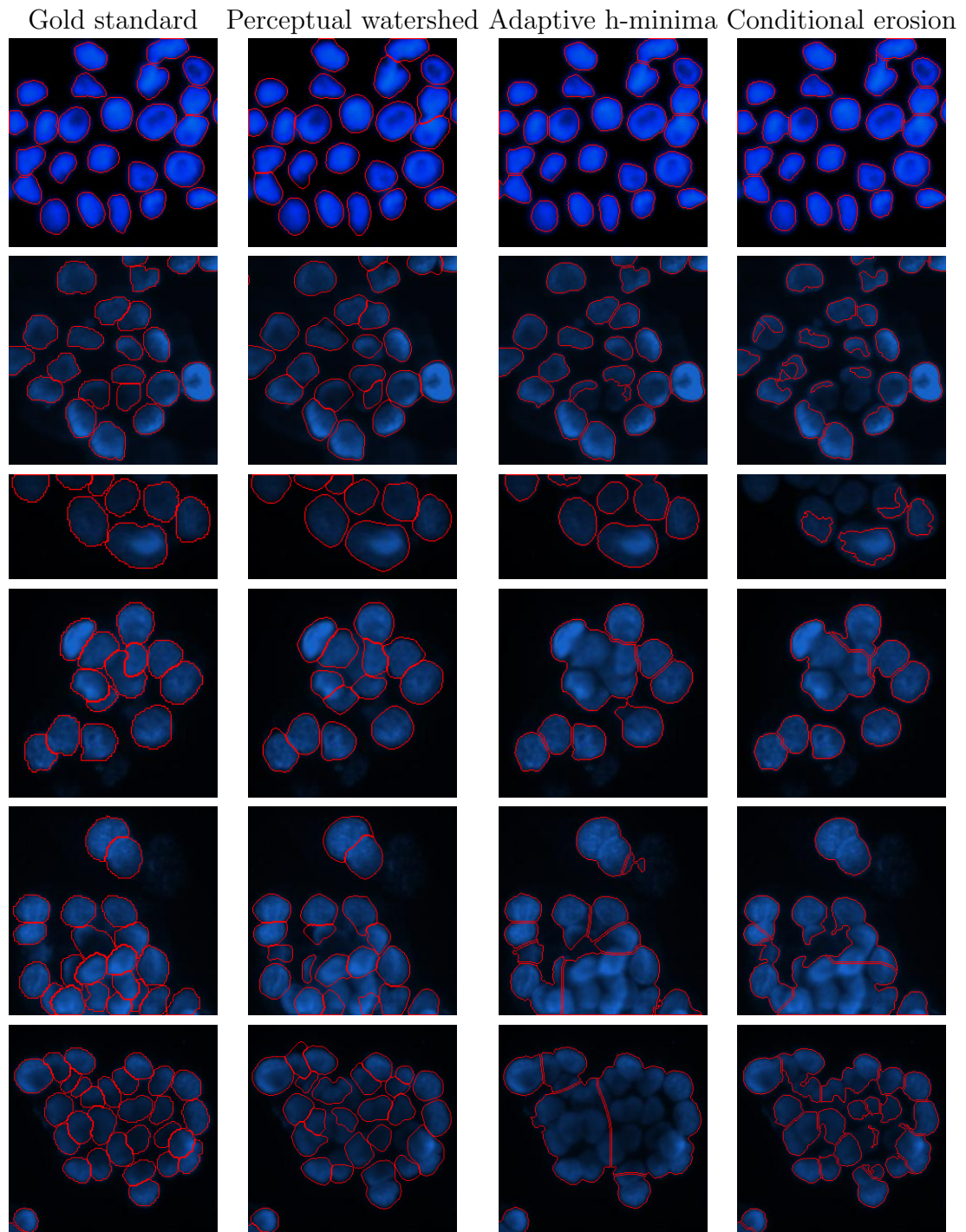


Figure 4.1: Visual results on example subimages obtained by the proposed perceptual watershed algorithm and the comparison methods. The size of the subimages is scaled for better visualization.

threshold levels but nowhere else, which makes it more robust to inaccuracies of this map.

Table 4.1: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of computed-annotated cell matches. The results are obtained on the Huh7 dataset.

	One-to-one	Overseg	Underseg	False	Miss
Perceptual watershed	1232	44	72	28	30
Adaptive h-minima	1140	14	205	45	19
Conditional erosion	1131	35	138	61	74

Table 4.2: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of computed-annotated cell matches. The results are obtained on the HepG2 dataset.

	One-to-one	Overseg	Underseg	False	Miss
Perceptual watershed	1002	53	153	62	75
Adaptive h-minima	867	4	390	51	22
Conditional erosion	820	19	364	63	80

Second and more importantly, our algorithm uses the fact that a cell should have four types of boundaries, each of which locates one of its four sides, in its segmentation. This helps our algorithm identify cells even when their boundaries are partially present (using the 4PRIM pattern) or even one of them is completely missing (using the 3PRIM pattern). This, in turn, is effective to compensate some negative effects of the confluency, since confluent cells commonly have boundaries partially or completely obscured by other cells. This property of our algorithm can also be seen in the visual results of the algorithms given in the last three rows of Fig. 4.1.

It is also worth to noting that we apply a post-process to the segmentation results of adaptive h -minima and conditional erosion. For this, we eliminate the segmented components that would not be labeled as cells due to their sizes, since, according to our observations, these components are spurious pixels that arise due to inadequate capabilities of the algorithms, which cannot sufficiently handle uneven illumination and noises. After this post-process, the precision measures for the algorithms have increased dramatically for both datasets. The precision measure of adaptive h -minima has increased from 68.23 to 80.20 percent

and from 82.19 to 87.90 percent for the HepG2 and Huh7 dataset, respectively. Similarly, the precision measure of conditional erosion has increased from 67.56 to 76.38 percent and from 77.38 to 88.17 percent for the HepG2 and Huh7 dataset, respectively. Additionally, especially for the HepG2 dataset, oversegmentations and false detections have also decreased by at least 2 or 3 times due to elimination of spurious pixels in and out of cellular regions. The huge change in the results is also a proof for the dependence of adaptive h -minima and conditional erosion to their initial segmentation masks.

Table 4.3: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the cell-based precision, recall, and F-score measures. The results are obtained on the Huh7 dataset.

	Precision	Recall	F-score
Perceptual watershed	89.15	89.40	89.28
Adaptive h-minima	87.90	82.73	85.23
Conditional erosion	85.17	82.08	83.59

Table 4.4: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the cell-based precision, recall, and F-score measures. The results are obtained on the HepG2 dataset.

	Precision	Recall	F-score
Perceptual watershed	80.22	78.10	79.15
Adaptive h-minima	80.20	67.58	73.35
Conditional erosion	76.35	63.91	69.58

Table 4.5: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the pixel-based precision, recall, and F-score measures. The results are obtained on the Huh7 dataset.

	Precision	Recall	F-score
Perceptual watershed	77.87	86.11	81.78
Adaptive h-minima	79.97	77.35	78.64
Conditional erosion	84.06	71.20	77.10

Table 4.6: Comparison of the proposed perceptual watershed algorithm against the previous methods in terms of the pixel-based precision, recall, and F-score measures. The results are obtained on the HepG2 dataset.

	Precision	Recall	F-score
Perceptual watershed	65.22	74.54	69.57
Adaptive h-minima	65.45	65.10	65.27
Conditional erosion	66.15	57.42	61.48

4.6 Parameter Analysis

In this section, we investigate the effects of each parameter to the segmentation performance of the proposed method. To this end, we fix three of the four parameters and observe the changes in the F-score, precision, and recall percentages with respect to different values of the remaining parameter. In Figs. 4.2 - Fig. 4.4, F-score, precision, and recall measures are presented for each parameter, respectively. The first parameter is the primitive length threshold t_{size} , which is used in both primitive definition and graph construction. In primitive definition, small primitive components whose width/height is smaller than this threshold are eliminated as they are likely to be noise rather than cell boundaries. In graph construction, because of the same reason, primitives smaller than t_{size} are not counted as the part of a structural pattern. Increasing this parameter results in eliminating more primitives, especially the ones that belong to small cells. This increases misses as well as undersegmentations. On the other hand, selecting a small threshold increases the amount of false primitives, which results in more false detections and oversegmentations. According to the size of the cells and the resolution of the images, we repeated our experiments by selecting the size threshold as $\{5, 10, 15, 20, 25, 30\}$. After the experiments, t_{size} is selected as 15 pixels. As observed in Fig. 4.2(a), this selected value gives good F-scores for both of the datasets.

The second parameter is the percentage threshold t_{perc} used by the iterative search algorithm. The search process relies on finding cell-like structural patterns in the graph G , constructed on primitives which are identified by thresholding Sobel responses. The initial values of threshold levels are selected using the

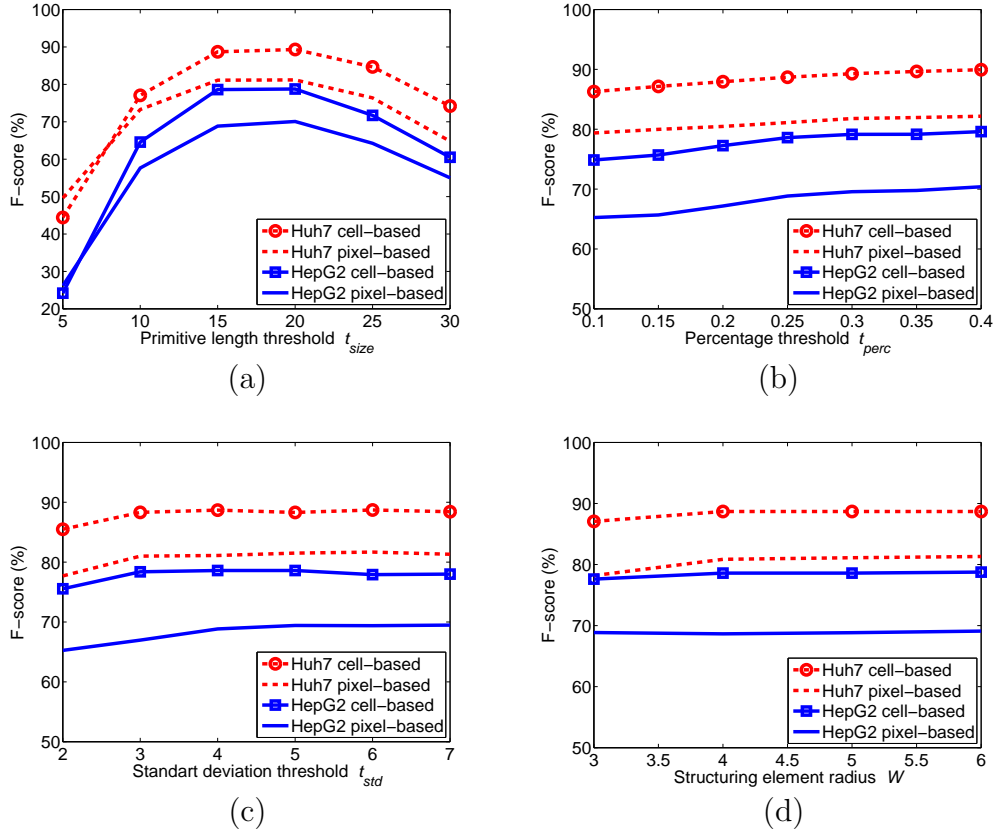


Figure 4.2: For the Huh7 and HepG2 datasets, cell-based and pixel-based F-score measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element.

Otsu method and they are relaxed at next iterations to find more primitives in the image. However, this relaxation should have a limit since the use of too small threshold levels, obtained after several iterations, falsely identifies noise as primitives. Thus, we use the percentage threshold t_{perc} to stop the iterations so that segmentation is not affected by these falsely identified primitives. Smaller values of this parameter lead to more false detections whereas larger values may fail to identify some true cells. With selection of the percentage threshold as $\{0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40\}$, the F-score results for both of the datasets are given in Fig. 4.2(b).

The third parameter is the standard deviation threshold t_{std} used in cell localization. In the iterative search algorithm, structural patterns are identified as

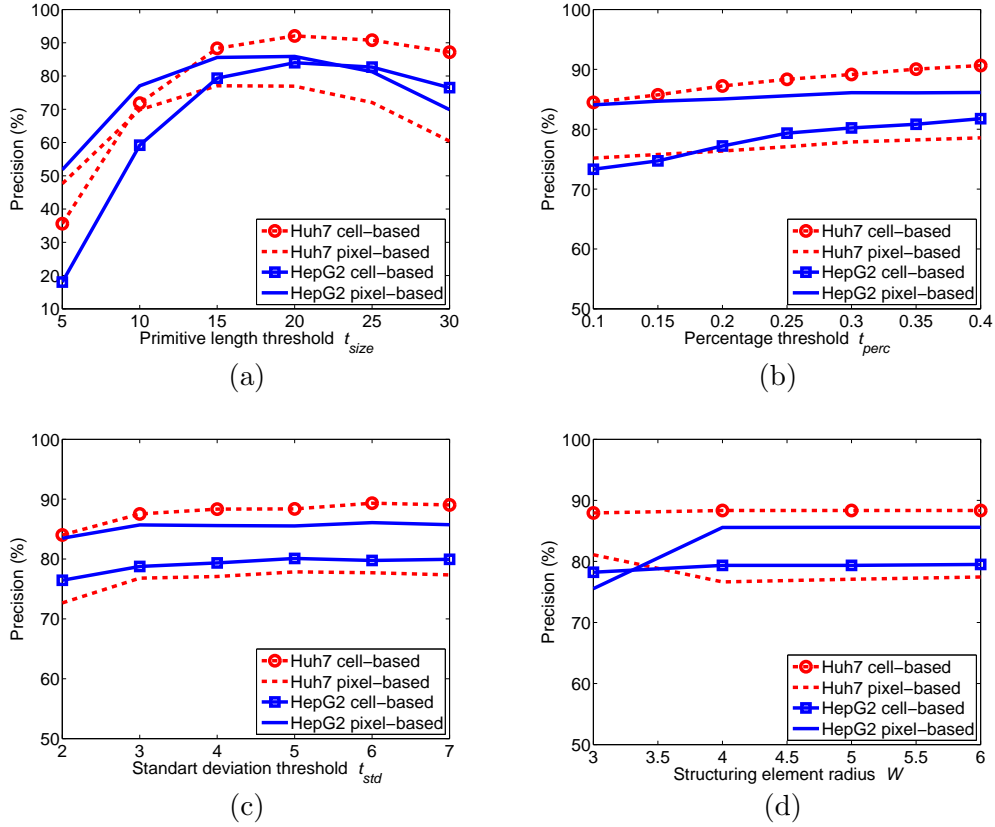


Figure 4.3: For the Huh7 and HepG2 datasets, cell-based and pixel-based precision measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element.

cells if they satisfy the shape constraint, which is based on the standard deviation of the cells. More regular-shaped cells are processed in first iterations whereas more irregular ones are dealt with later. Thus, an initial value t_{std} is used for this parameter and it is increased in each iteration to process irregular cells as well. By starting with smaller threshold values, the algorithm processes a smaller number of regular shapes in its first iterations and treats regular and irregular shapes the same in later iterations. On the other hand, starting with larger values, a greater number of irregular cells, which may correspond to multiple regular cells, are identified in first iterations. With selection of the percentage threshold as $\{3, 4, 5, 6, 7\}$, the analysis results are shown in Fig. 4.2(c). As seen in the figure, both of them slightly lower the segmentation performance.

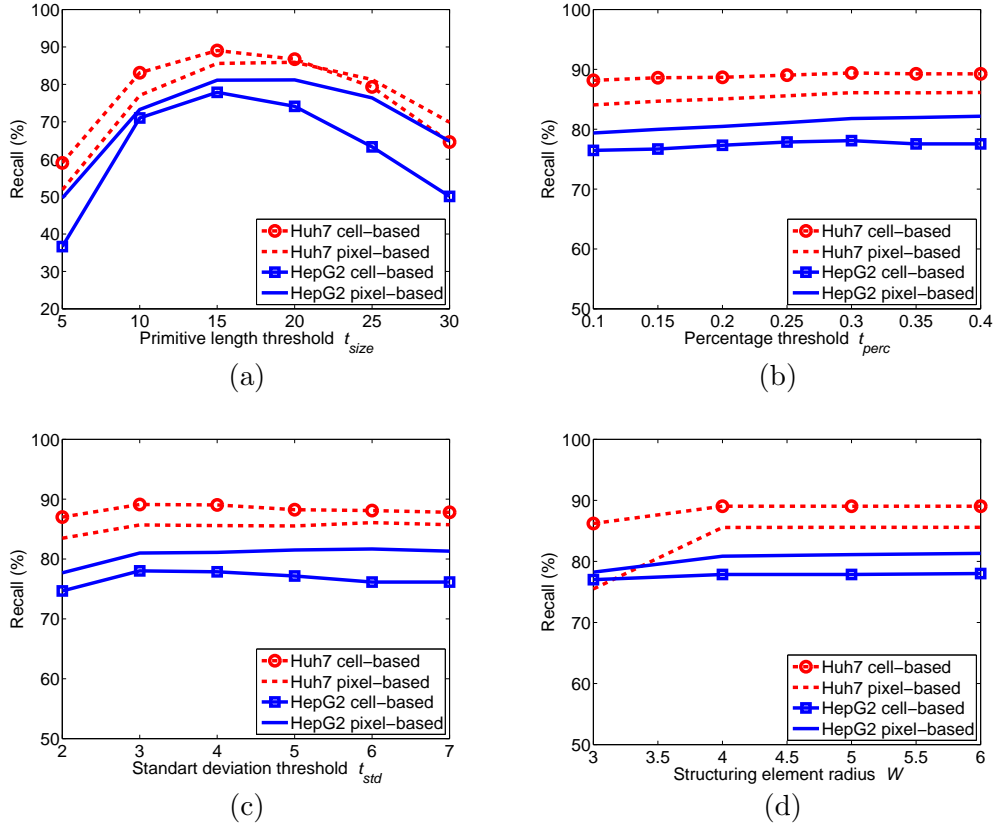


Figure 4.4: For the Huh7 and HepG2 datasets, cell-based and pixel-based recall measures as a function of (a) the primitive length threshold t_{size} , (b) the percentage threshold t_{perc} , (c) the standard deviation threshold t_{std} , and (d) the radius W of the structuring element.

The last parameter is the radius of the structural element W , which is used in region growing. After obtaining the segmentation results, we apply majority filtering, with a disk structuring element of radius W , to smooth the boundaries. Different values of W do not affect the performance of the segmentation algorithm much. But selection of a very small radius yields cells with rough boundaries, which increases oversegmentations. To quantitatively understand the effect of this parameter, we repeated our experiments with selecting the radius of structuring element as $\{3, 4, 5, 6\}$. The analysis results (Fig. 4.2(d)) show that selection of this parameter affects the segmentation performance less compared to other parameters. On the other hand, the recall measures for W , given in Fig. 4.4(d), prove that increasing W slightly yields better cell areas, but further increasing this parameter does not affect the results much.

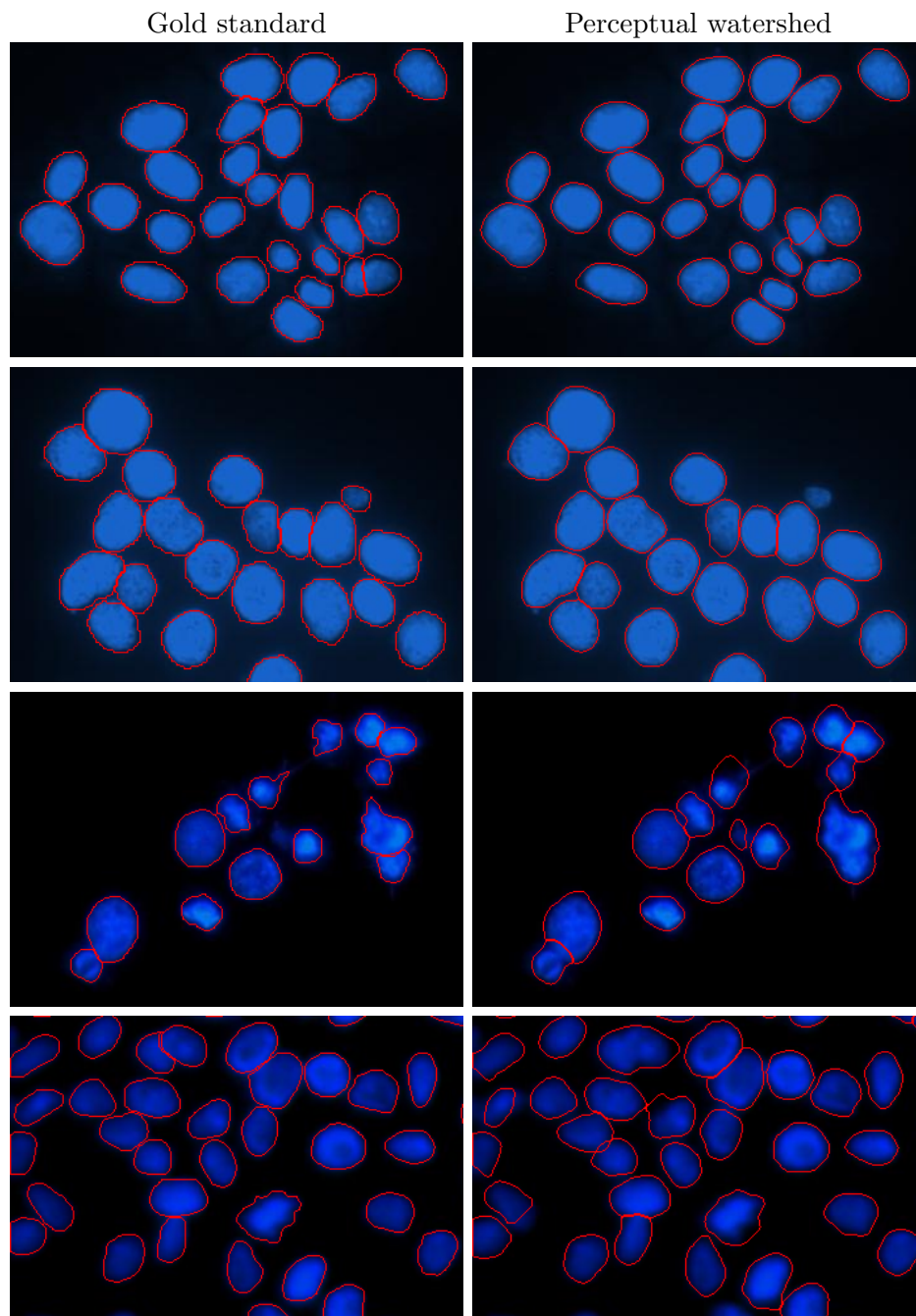


Figure 4.5: Visual results on example subimages obtained by the proposed perceptual watershed algorithm.

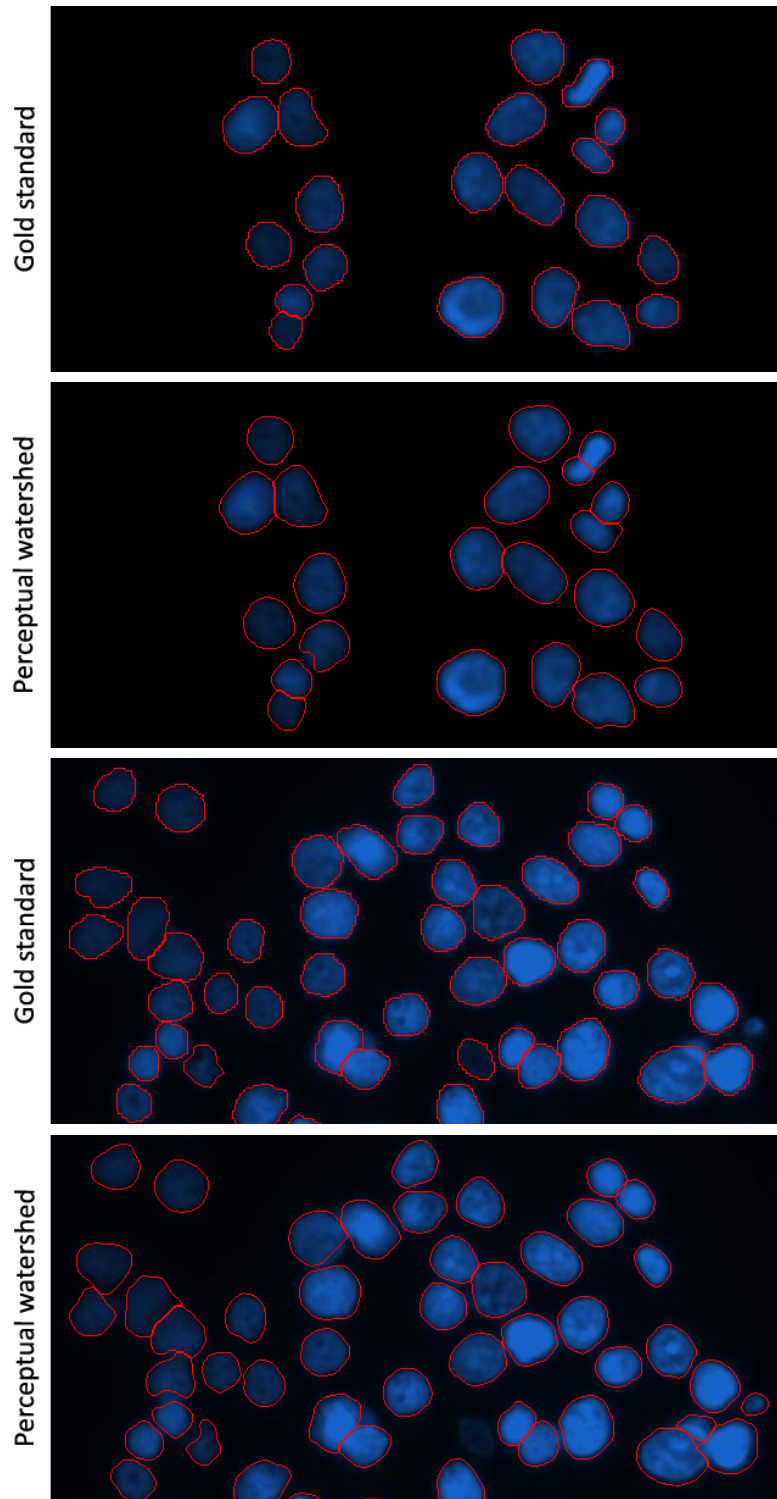


Figure 4.6: Visual results on example subimages obtained by the proposed perceptual watershed algorithm.

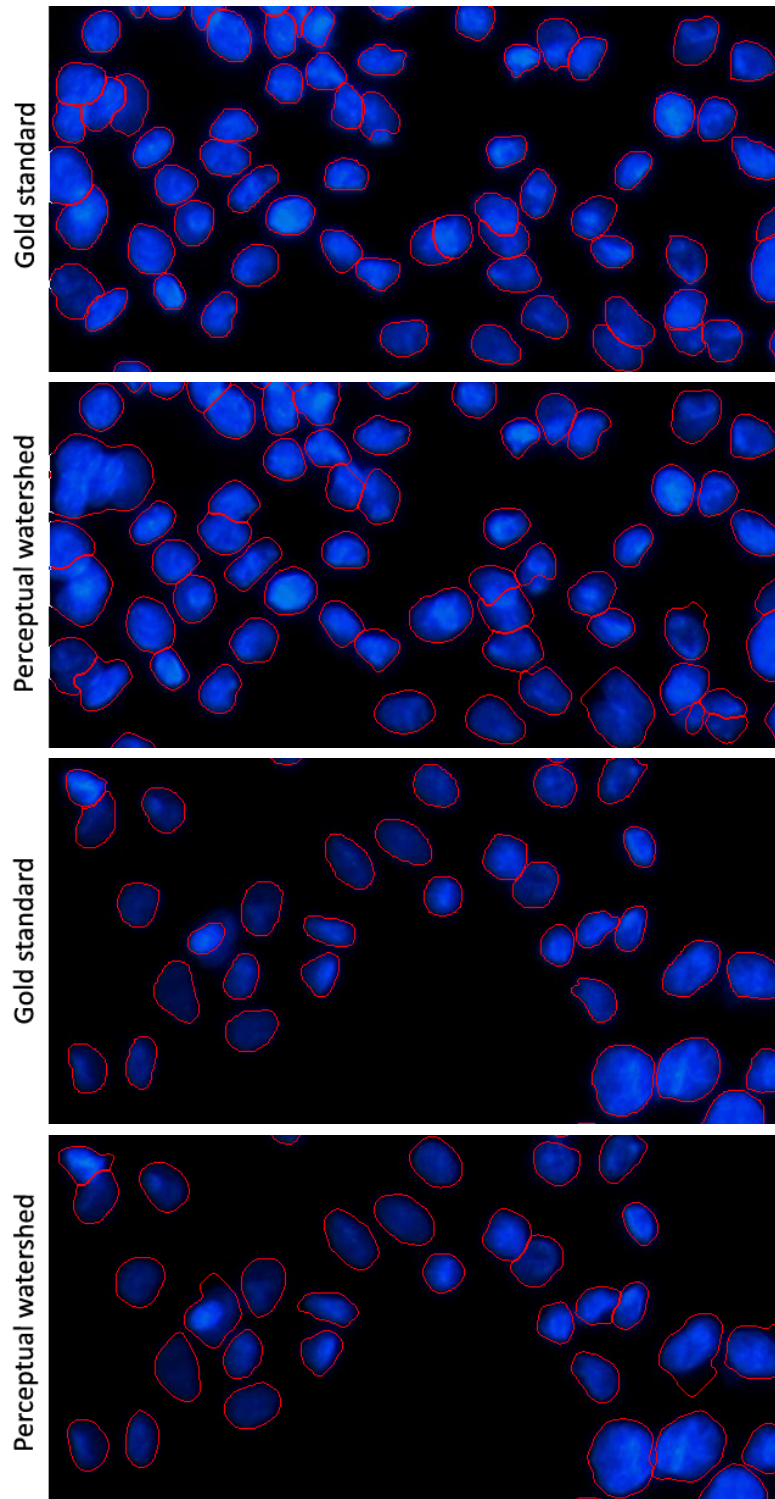


Figure 4.7: Visual results on example subimages obtained by the proposed perceptual watershed algorithm.

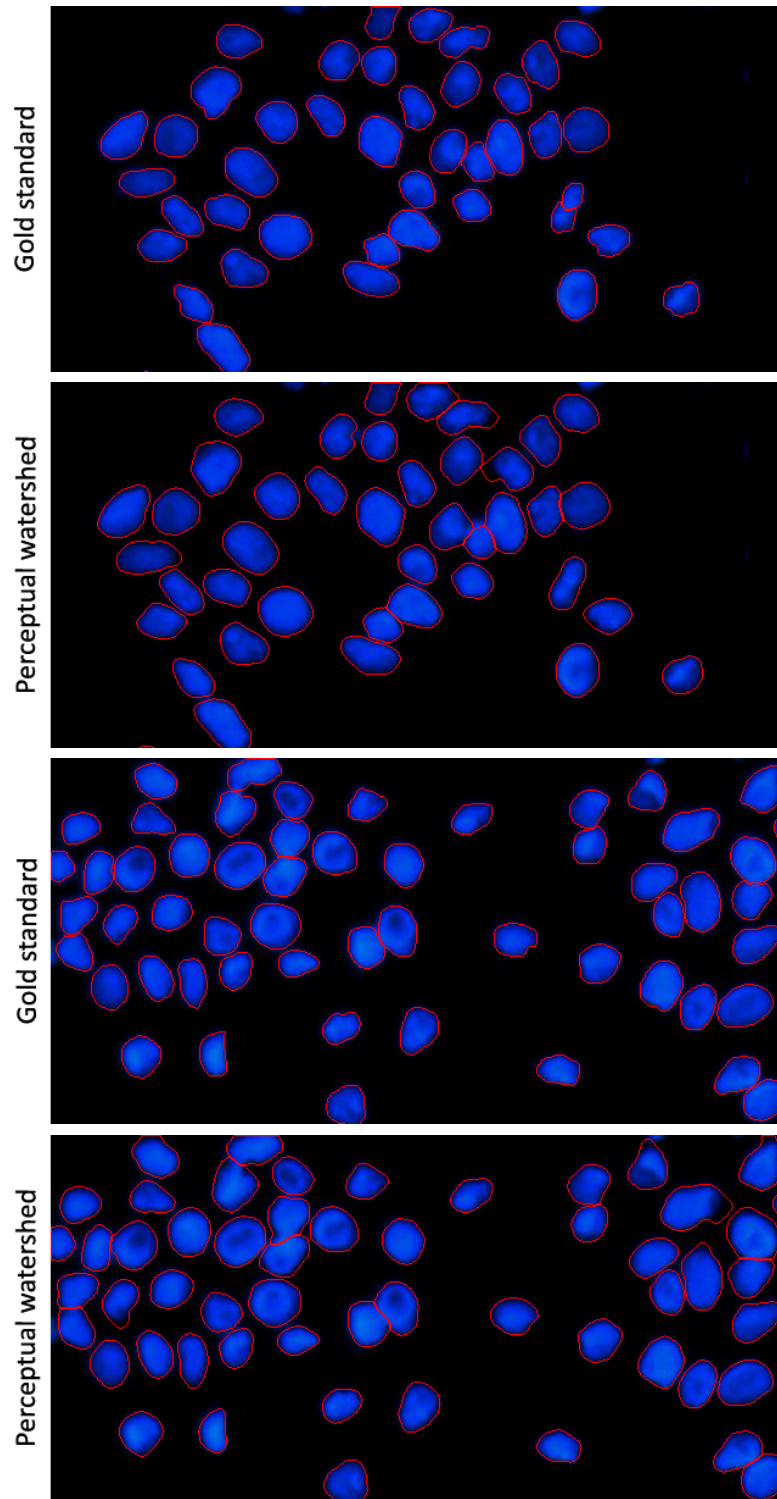


Figure 4.8: Visual results on example subimages obtained by the proposed perceptual watershed algorithm.

Chapter 5

Conclusion

This thesis presents a new marker-controlled watershed algorithm for segmenting cells in fluorescence microscopy images. This algorithm proposes to incorporate human perception, in the form of how a human locates a cell by identifying the boundaries on its four sides and joining these boundaries, into segmentation. To this end, it defines high-level primitives to represent the boundaries and transforms cell segmentation into a problem of identifying predefined structural patterns in the graph constructed on these boundary primitives. It then considers the identified patterns as its markers and grows these markers by making use of the primitive locations. The proposed algorithm is tested on 2661 images of two different cell lines. The experiments reveal that the proposed algorithm leads to better results for both less confluent and more confluent cells, compared to its previous counterparts.

One future research direction is to define a marking function based on the primitive-specific distances. Thus, one could guide the marker growing process by using this marking function. Since the proposed method identifies markers with a high accuracy, implementation of such marking function is expected to yield better segmentation boundaries. Besides, definition of a primitive-based distance transform and marking function would be an interesting research problem.

In different applications of molecular cellular biology research, it is very important to understand and compare cell morphologies, which can be expressed using quantitative measures. As previously mentioned, cell segmentation is the first step to derive such measures. As another future research direction, one could work on extracting these measures after segmenting cells by the proposed algorithm. Additionally, when cells are co-stained with different agents to understand different molecular mechanisms, determining cell boundaries may help obtain better sensitivity and specificity of measuring these agents. Thus, it would be an interesting future research direction to investigate the use of the proposed algorithm for these cells and to make modifications, if necessary.

Bibliography

- [1] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [2] N. Thomas, “High-content screening: A decade of evolution,” *J Biomol Screen*, vol. 15, no. 1, pp. 1–9, 2010.
- [3] X. Chen, X. Zhou, and S. T. C. Wong, “Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 4, pp. 762–766, 2006.
- [4] N. Harder, F. Mora-Bermudez, W. J. Godinez, J. Ellenberg, R. Eils, and K. Rohr, “Automated analysis of the mitotic phases of human cells in 3D fluorescence microscopy image sequences,” *Med Image Comput Comput Assist Interv*, vol. 9, no. 1, pp. 840–848, 2006.
- [5] D. Fenistein, B. Lenseigne, T. Christophe, P. Brodin, and A. Genovesio, “A fast, fully automated cell segmentation algorithm for high-throughput and high-content screening,” *Cytometry Part A*, vol. 73A, no. 10, pp. 958–964, 2008.
- [6] X. Zhou, F. Li, J. Yan, and S. T. C. Wong, “A novel cell segmentation method and cell phase identification using Markov model,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 13, no. 2, pp. 152–157, 2009.
- [7] J. Cheng and J. Rajapakse, “Segmentation of clustered nuclei with shape

- markers and marking function,” *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 741–748, 2009.
- [8] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, and S. T. C. Wong, “Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy,” *Bioinformatics*, vol. 24, no. 1, pp. 94–101, 2008.
- [9] X. Yang, H. Li, and X. Zhou, “Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and Kalman filter in time-lapse microscopy,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2405–2414, 2006.
- [10] P. S. U. Adiga and B. B. Chaudhuri, “An efficient method based on watershed and rule-based merging for segmentation of 3-D histo-pathological images,” *Pattern Recognition*, vol. 34, no. 7, pp. 1449–1458, 2001.
- [11] C. Wählby, I.-M. Sintorn, F. Erlandsson, G. Borgefors, and E. Bengtsson, “Combining intensity, edge and shape information for 2D and 3D segmentation of cell nuclei in tissue sections,” *Journal of Microscopy*, vol. 215, no. 1, pp. 67–76, 2004.
- [12] “The Fluorescence Microscope.” <http://www.nobelprize.org/educational/physics/microscopes/fluorescence/index.html>. Accessed in July 2012.
- [13] X. Zhou and S. T. C. Wong, “High content cellular imaging for drug development,” *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 170–174, 2006.
- [14] R. Pepperkok and J. Ellenberg, “High-throughput fluorescence microscopy for systems biology,” *Nat Rev Mol Cell Biol*, vol. 7, no. 9, pp. 690–696, 2006.
- [15] I. N. Bankman, ed., *Handbook of Medical Image Processing and Analysis*. Academic Press, 2nd ed., 2008.
- [16] F. Zanella, J. B. Lorens, and W. Link, “High content screening: seeing is believing,” *Trends in Biotechnology*, vol. 28, no. 5, pp. 237–245, 2010.

- [17] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2001.
- [18] S.-T. Bow, *Pattern Recognition and Image Preprocessing*. New York, NY, USA: Marcel Dekker, Inc., 2nd ed., 2002.
- [19] C. Wählby, J. Lindblad, M. Vondrus, E. Bengtsson, and L. Bjorkestén, “Algorithms for cytoplasm segmentation of fluorescence labelled cells,” *Anal Cell Pathol*, vol. 24, no. 2-3, pp. 101–111, 2002.
- [20] G. Xiong, X. Zhou, and L. Ji, “Automated segmentation of drosophila RNAi fluorescence cellular images using deformable models,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 11, pp. 2415–2424, 2006.
- [21] M. E. Plissiti, C. Nikou, and A. Charchanti, “Combining shape, texture and intensity features for cell nuclei extraction in pap smear images,” *Pattern Recognition Letters*, vol. 32, no. 6, pp. 838–853, 2011.
- [22] K. Zuiderveld, “Graphics gems iv,” ch. Contrast limited adaptive histogram equalization, pp. 474–485, San Diego, CA, USA: Academic Press Professional, Inc., 1994.
- [23] P. Quelhas, M. Marcuzzo, A. Mendonca, and A. Campilho, “Cell nuclei and cytoplasm joint segmentation using the sliding band filter,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 8, pp. 1463–1473, 2010.
- [24] X. Xu, Z. Yang, and Y. Wang, “A method based on rank-ordered filter to detect edges in cellular image,” *Pattern Recognition Letters*, vol. 30, no. 6, pp. 634–640, 2009.
- [25] G. Lin, M. K. Chawla, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, “Hierarchical, model-based merging of multiple fragments for improved three-dimensional segmentation of nuclei,” *Cytometry Part A*, vol. 63A, no. 1, pp. 20–33, 2005.
- [26] G. Lin, U. Adiga, K. Olson, J. F. Guzowski, C. A. Barnes, and B. Roysam, “A hybrid 3D watershed algorithm incorporating gradient cues and object

- models for automatic segmentation of nuclei in confocal image stacks,” *Cytometry Part A*, vol. 56A, no. 1, pp. 23–36, 2003.
- [27] N. Kharma, H. Moghnieh, J. Yao, Y. P. Guo, A. Abu-Baker, J. Laganieri, G. Rouleau, and M. Cheriet, “Automatic segmentation of cells from microscopic imagery using ellipse detection,” *IET Image Processing*, vol. 1, no. 1, pp. 39–47, 2007.
- [28] K. Fu and J. Mui, “A survey on image segmentation,” *Pattern Recognition*, vol. 13, no. 1, pp. 3–16, 1981.
- [29] D. L. Pham, C. Xu, and J. L. Prince, “Current methods in medical image segmentation,” *Annual Review of Biomedical Engineering*, vol. 2, no. 1, pp. 315–337, 2000.
- [30] S. Yanowitz and A. Bruckstein, “A new method for image segmentation,” in *Proceedings of the 9th International Conference on Pattern Recognition*, pp. 270–275, 1988.
- [31] M. Sezgin and B. Sankur, “Survey over image thresholding techniques and quantitative performance evaluation,” *Journal of Electronic Imaging*, vol. 13, no. 1, pp. 146–168, 2004.
- [32] B. Ballaró, A. M. Florena, V. Franco, D. Tegolo, C. Tripodo, and C. Valenti, “An automated image analysis methodology for classifying megakaryocytes in chronic myeloproliferative disorders,” *Medical Image Analysis*, vol. 12, no. 6, pp. 703–712, 2008.
- [33] B. Fang, W. Hsu, and M. L. Lee, “On the accurate counting of tumor cells,” *IEEE Transactions on NanoBioscience*, vol. 2, no. 2, pp. 94–103, 2003.
- [34] H.-S. Wu, J. Berba, and J. Gil, “Iterative thresholding for segmentation of cells from noisy images,” *Journal of Microscopy*, vol. 197, no. 3, pp. 296–304, 2000.
- [35] X. Chen and S. T. C. Wong, “Automated dynamic cellular analysis in high throughput drug screens,” in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 4229–4232, 2005.

- [36] N. Malpica, C. O. de Solórzano, J. J. Vaquero, A. Santos, I. Vallcorba, J. M. García-Sagredo, and F. del Pozo, “Applying watershed algorithms to the segmentation of clustered nuclei,” *Cytometry*, vol. 28, no. 4, pp. 289–297, 1997.
- [37] J. Lindblad, C. Wählby, L. Ji, E. Bengtsson, and A. Zaltsman, “Image analysis for automatic segmentation of cytoplasm and classification of rac1 activation,” *Cytometry Part A*, vol. 57A, pp. 22–33, 2004.
- [38] P. R. Gudla, K. Nandy, J. Collins, K. J. Meaburn, T. Misteli, and S. J. Lockett, “A high-throughput system for segmenting nuclei using multiscale techniques,” *Cytometry Part A*, vol. 73A, no. 5, pp. 451–466, 2008.
- [39] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, “Elastically deformable models,” *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 205–214, 1987.
- [40] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International Journal of Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [41] P. Moore and D. Molloy, “A survey of computer-based deformable models,” in *Proceedings of the International Machine Vision and Image Processing Conference*, pp. 55–66, 2007.
- [42] C. Xu and J. Prince, “Gradient vector flow: a new external force for snakes,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 66–71, 1997.
- [43] T. Chan and L. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.
- [44] C. Zimmer, E. Labruyere, V. Meas-Yedid, N. Guillen, and J.-C. Olivo-Marin, “Segmentation and tracking of migrating cells in videomicroscopy with parametric active contours: a tool for cell-based drug testing,” *IEEE Transactions on Medical Imaging*, vol. 21, no. 10, pp. 1212–1221, 2002.
- [45] C. Doukas and I. Maglogiannis, “Automated cell apoptosis characterization

- using active contours,” in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 812–815, 2007.
- [46] N. Ray and S. Acton, “Tracking fast-rolling leukocytes in vivo with active contours,” in *Proceedings of the International Conference on Image Processing*, vol. 3, pp. III.165–III.168, 2002.
- [47] F. Bunyak, K. Palaniappan, S. Nath, T. Baskin, and G. Dong, “Quantitative cell motility for in vitro wound healing using level set-based active contour tracking,” in *Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: Nano to Macro*, pp. 1040–1043, 2006.
- [48] L. A. Vese and T. F. Chan, “A multiphase level set framework for image segmentation using the mumford and shah model,” *Int. J. Comput. Vision*, vol. 50, no. 3, pp. 271–293, 2002.
- [49] B. Zhang, C. Zimmer, and J.-C. Olivo-Marin, “Tracking fluorescent cells with coupled geometric active contours,” in *Proceedings of the IEEE International Symposium on Biomedical Imaging: Nano to Macro*, vol. 1, pp. 476–479, 2004.
- [50] C. Zimmer and J.-C. Olivo-Marin, “Coupled parametric active contours,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1838–1842, 2005.
- [51] A. Dufour, V. Shinin, S. Tajbakhsh, N. Guillen-Aghion, J.-C. Olivo-Marin, and C. Zimmer, “Segmenting and tracking fluorescent cells in dynamic 3-D microscopy with coupled active surfaces,” *IEEE Transactions on Image Processing*, vol. 14, no. 9, pp. 1396–1410, 2005.
- [52] F. Meyer and S. Beucher, “Morphological segmentation,” *Journal of Visual Communication and Image Representation*, vol. 1, no. 1, pp. 21–46, 1990.
- [53] L. Vincent and P. Soille, “Watersheds in digital spaces: An efficient algorithm based on immersion simulations,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, 1991.

- [54] S. Beucher and F. Meyer, “The morphological approach to segmentation: the watershed transformation. Mathematical morphology in image processing,” *Optical Engineering*, vol. 34, pp. 433–481, 1993.
- [55] P. Soille, *Morphological Image Analysis: Principles and Applications*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2nd ed., 2003.
- [56] C. Jung and C. Kim, “Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 10, pp. 2600–2604, 2010.
- [57] M. Fatima and V. Seenivasagam, “A fast fuzzy-c means based marker controlled watershed segmentation of clustered nuclei,” in *Proceedings of the International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pp. 186–192, 2011.
- [58] K. Mao, P. Zhao, and P.-H. Tan, “Supervised learning-based cell image segmentation for p53 immunohistochemistry,” *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 6, pp. 1153–1163, 2006.
- [59] H. Zhou and K. Mao, “Adaptive successive erosion-based cell image segmentation for p53 immunohistochemistry in bladder inverted papilloma,” in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society*, pp. 6484–6487, 2005.
- [60] A. Pinidiyaarachchi and C. Wählby, “Seeded watersheds for combined segmentation and tracking of cells,” in *Proceedings of the Image Analysis and Processing*, pp. 336–343, 2005.
- [61] C. Jung, C. Kim, S. W. Chae, and S. Oh, “Unsupervised segmentation of overlapped nuclei using Bayesian classification,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 12, pp. 2825–2832, 2010.
- [62] A. Bleau and L. J. Leon, “Watershed-based segmentation and region merging,” *Comput Vis Image Underst*, vol. 77, no. 3, pp. 317–370, 2000.

- [63] T. Jiang, F. Yang, Y. Fan, and D. J. Evans, “A parallel genetic algorithm for cell image segmentation,” *Electronic Notes in Theoretical Computer Science*, vol. 46, pp. 214–224, 2001.
- [64] H. Kong, M. Gurcan, and K. Belkacem-Boussaid, “Partitioning histopathological images: An integrated framework for supervised color-texture segmentation and cell splitting,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 9, pp. 1661–1677, 2011.
- [65] S. Kumar, S. Ong, S. Ranganath, T. Ong, and F. Chew, “A rule-based approach for robust clump splitting,” *Pattern Recognition*, vol. 39, no. 6, pp. 1088–1098, 2006.
- [66] A. Mouelhi, M. Sayadi, and F. Fnaiech, “Automatic segmentation of clustered breast cancer cells using watershed and concave vertex graph,” in *Proceedings of the International Conference on Communications, Computing and Control Applications*, pp. 1–6, 2011.
- [67] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam, “Improved automatic detection and segmentation of cell nuclei in histopathology images,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 4, pp. 841–852, 2010.
- [68] C. Chen, H. Li, X. Zhou, and S. Wong, “Graph cut based active contour for automated cellular image segmentation in high throughput RNA interference (RNAi) screening,” in *Proceedings of the 4th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 69–72, 2007.
- [69] O. Daněk, 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*, pp. 410–419, 2009.
- [70] F. Li, X. Zhou, J. Ma, and S. Wong, “Multiple nuclei tracking using integer programming for quantitative cancer cell cycle analysis,” *IEEE Transactions on Medical Imaging*, vol. 29, no. 1, pp. 96–105, 2010.

- [71] D. Padfield, J. Rittscher, and B. Roysam, “Coupled minimum-cost flow cell tracking for high-throughput quantitative analysis,” *Medical Image Analysis*, vol. 15, no. 4, pp. 650–668, 2011.
- [72] F. Yang, M. A. Mackey, F. Ianzini, G. Gallardo, and M. Sonka, “Cell segmentation, tracking, and mitosis detection using temporal context,” *Med Image Comput Comput Assist Interv*, vol. 8, no. 1, pp. 302–309, 2005.