

SMART MARKERS FOR WATERSHED-BASED CELL SEGMENTATION

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

Can Fahrettin Koyuncu

September, 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.

Assist. Prof. Dr. Can Alkan

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. Özlen Konu

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

SMART MARKERS FOR WATERSHED-BASED CELL SEGMENTATION

Can Fahrettin Koyuncu
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. Çiğdem Gündüz Demir
September, 2012

Automated cell imaging systems facilitate fast and reliable analysis of biological events at the cellular level. In these systems, the first step is usually cell segmentation that greatly affects the success of the subsequent system steps. On the other hand, similar to other image segmentation problems, cell segmentation is an ill-posed problem that typically necessitates the use of domain specific knowledge to obtain successful segmentations even by human subjects. The approaches that can incorporate this knowledge into their segmentation algorithms have a potential to greatly improve the segmentation results.

In this study, we propose a new approach for the effective segmentation of live cells from phase-contrast microscopy. This approach introduces a new set of “smart markers” for a marker-controlled watershed algorithm, for which the identification of its markers is critical. The proposed approach relies on using domain specific knowledge, in the form of visual characteristics of the cells, to define the markers. We evaluate our approach on a total of 1954 cells. The experimental results demonstrate that the proposed approach is quite effective in identifying better markers compared to its counterparts. This will in turn be effective in improving the segmentation performance of a marker-controlled watershed algorithm.

Keywords: Marker-controlled watersheds, live cell segmentation, phase contrast microscopy images, cell lines.

ÖZET

SU-SEDDİ ALGORİTMASI TABANLI HÜCRE BÖLÜTLEMESİ İÇİN AKILLI İŞARETÇİ TESPİTİ

Can Fahrettin Koyuncu

Bilgisayar Mühendisliği, Yüksek Lisans

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

Eylül, 2012

Otomatik hücre görüntüleme sistemleri, hücre seviyesindeki biyolojik olayların hızlı ve güvenilir bir şekilde analizine olanak sağlar. Bu sistemlerde genellikle ilk basamak, daha sonraki sistem basamaklarını büyük ölçüde etkileyen hücre bölütlemesidir. Öte yandan, diğer görüntü bölütleme problemlerine benzer şekilde, hücre bölütleme eksik tanımlanmış bir problemdir. Bu problem, tipik olarak, insanların başarılı bölütlemeler elde edebilmesi için bile, alan bilgisinin kullanılmasını gerektirir. Bu bilgiyi bölütleme algoritmalarına dahil eden yaklaşımların, bölütleme sonuçlarını büyük ölçüde iyileştirme potansiyelleri vardır.

Biz bu çalışmamızda, faz-kontrast mikroskobu ile alınan canlı hücrelerin etkin bölütlenmesi için yeni bir yaklaşım önermekteyiz. Bu yaklaşım, işaretçilerinin tespiti için kritik olduğu işaretçi-kontrollü su-seddi algoritmaları için yeni bir “akıllı işaretçi” kümesi tanımlar. Önerilen yaklaşım, hücrelerin görsel karakteristikleri biçiminde ifade edilen alana özel bilginin işaretçi tanımında kullanılmasına dayanır. Bu yaklaşım, toplam 1954 hücre üzerinde değerlendirilmiştir. Deneysel çalışmalar, önerilen yaklaşımın, diğer yaklaşımlarla karşılaştırıldığında daha iyi işaretçiler bulunmasında etkin olduğunu göstermiştir. Bu ise, işaretçi-kontrollü su-seddi algoritmalarının bölütleme performanslarının iyileştirilmesinde etkili olacaktır.

Anahtar sözcükler: İşaretçi-kontrollü su-seddi, canlı hücre bölütlemesi, faz kontrast mikroskop resimleri, hücre hatları.

Acknowledgement

This thesis is the end of my journey in obtaining my M.S degree. At the end of this journey, I would like to thank all people, who made this thesis possible.

My first debt of gratitude must go to my advisor, Assist. Prof. Dr. iğdem Gündüz Demir. Without her guidance and help, I would not finish this thesis. I would also like to thank to Assist. Prof. Dr. Özlen Konu and Assist. Prof. Dr. Can Alkan for reviewing and commenting on this thesis.

I would like to thank to the Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing financial assistance during my study, through the project TÜBİTAK 109E206 and the BİDEB program.

My friends were sources of laughter, joy, and support. Sincere thanks to all my friends especially Erdem, Oğuz, Enver, Barış, Shatlyk, Burak and others for their kindness and moral support during my study. Special thanks go to Mustafa, Zeynep, İlker, and Kıvanç for sharing their life with me. Their informal support and encouragement has been indispensable. I feel very lucky to have a valuable research partner, Salim. I could not complete my work without his invaluable and friendly assistance.

I owe my deepest gratitude to my parents, Necla Koyuncu and Ahmet Dursun Koyuncu, and my sister, İrem Özçifçi, and her husband, Gökçen Özçifçi. Their love, support, sacrifice and encouragement provided my inspiration and were my driving force. I owe them everything and wish I could show them just how much I love and appreciate them.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Contribution	4
1.3	Outline	4
2	Background	6
2.1	Automated Cell Segmentation	6
2.1.1	Thresholding	7
2.1.2	Active Contour Models	9
2.1.3	Watersheds	12
2.1.4	Other Methods	14
3	Methodology	15
3.1	Smart Markers Algorithm	15
3.1.1	Cell Pixel Quantization	16
3.1.2	Smart Marker Extraction	22

4	Experimental Results	27
4.1	Dataset	27
4.2	Parameter Selection	32
4.3	Comparisons	33
4.4	Results	34
4.5	Parameter Analysis	43
5	Conclusion	47

List of Figures

1.1	Example images of live KATO-3 gastric carcinoma cells. As shown in these images, the cells can be grouped into four morphological classes based on their visual characteristics. Examples from these groups are also indicated on the images.	3
2.1	(a) isolated cells, (b) clumped cells	7
2.2	Illustration for thresholding on intensity and gradient values. (a) original KATO-3 image, (b) its gray-level image, (c) binary image after applying Otsu thresholding on the gray image, (d) another original KATO-3 image, (e) its gradient map obtained by Sobel operators, (f) binary image after applying Otsu thresholding on the gradient map.	8
2.3	(a) Original KATO-3 image, and (b) labeled image after applying k-means clustering on red, green, and blue values of pixels.	9
2.4	Segmentation results obtained using active contour models on an example live cell image. (a)-(c) using a smaller initial curve: after (a) 20 iterations, (b) 80 iterations, and (c) 200 iterations. (d)-(f) using a larger initial curve: after (d) 10 iterations, (e) 50 iterations, and (f) 500 iterations.	10

2.5	Illustration of snakes and region-based active contours on an isolated cell having boundary with lower gradients. (a)-(c) correspond to the snake algorithm, and (d)-(f) correspond to the region-based active contour model.	11
2.6	Results using region-based active contour models on an example live cell image: after (a) 1 iteration, (b) 100 iterations, (c) 200 iterations, (d) 300 iterations, (e) 400 iterations, and (d) 500 iterations.	12
2.7	Illustration of a watershed algorithm. (a) Original image, and (b) flooding process starting from the darker regions.	13
3.1	Schematic overview of the proposed algorithm.	16
3.2	Illustration of the majority filtering with a square structuring element (size = 4): (a) before and (b) after.	18
3.3	Illustration of the cell pixel quantization step on three exemplary images. (a) start with an original image, (b) identify <i>bright</i> and <i>dark</i> pixels using the intensity and gradient information, (c) remove narrow dark pixels around of bright pixels and then apply majority filtering, (d) identify some of the dark pixels as <i>dark-center</i> pixels, (e) eliminate noise, artifacts and small background regions. In this illustration, the bright, dark, and dark-center pixels are shown with pink, green, and blue, respectively.	19
3.4	Illustration of the iterative erosion algorithm on the <i>dark center</i> pixels: (a) before and (b) after.	22
3.5	Illustration of the iterative erosion algorithm on the <i>dark</i> pixels: (a) before and (b) after.	23

3.6	Illustration of effects of the boundaries: (a) <i>bright</i> pixels, (b) circle fit algorithm on all <i>bright</i> pixels and (c) circle fit algorithm on the remaining <i>bright</i> pixels after we exclude the dilated version of the previously found markers.	23
3.7	Illustration of the effect of the constraint: (a) <i>bright</i> pixels, (b) circle fit algorithm without employing the constraint and (c) circle fit algorithm with employing the constraint.	24
3.8	For two example images, (a) <i>smart</i> markers identified by the proposed algorithm, and (b) cell boundaries obtained after growing these smart markers by employing dark, dark-center, and bright pixels.	26
4.1	Example images obtained from different cell lines: (a)-(b) MFE-296, (c)-(d) MCF-7, (e)-(f) KATO-3, (g)-(h) HUH-7.	28
4.2	For the training set, the number of one-to-one matches as a function of the distance threshold value used in our distance-based evaluation algorithm.	34
4.3	Visual results of the marker-identification algorithms, obtained on example images. (a) Gold standards and the markers found by (b) smart markers, (c) intensity-based, (d) distance-based, and (c) conditional-erosion algorithms.	39
4.4	Visual results of the marker-identification algorithms, obtained on another set of example images. (a) Gold standards and the markers found by (b) smart markers, (c) intensity-based, (d) distance-based, and (c) conditional-erosion algorithms.	40
4.5	Visual results of the watershed algorithms, after growing the markers, obtained on example images. (a) Gold standards and the boundaries found by (b) smart markers, (c) intensity-based, (d) distance-based, and (c) conditional-erosion algorithms.	41

4.6	Visual results of the watershed algorithms, after growing the markers, obtained on another set of example images. (a) Gold standards and the boundaries found by (b) smart markers, (c) intensity-based, (d) distance-based, and (e) conditional-erosion algorithms.	42
4.7	For the training set, the precision, recall, and F-score measures as a function of (a) the Sobel threshold constant k , (b) the size W of the majority filter, (c) the area threshold T_{area} , (d) the size threshold T_{size} , and (e) the radius r_{disk} of the structuring element.	45
4.8	For the test set, the precision, recall, and F-score measures as a function of (a) the Sobel threshold constant k , (b) the size W of the majority filter, (c) the area threshold T_{area} , (d) the size threshold T_{size} , and (e) the radius r_{disk} of the structuring element.	46
5.1	Visual results of the proposed algorithm obtained on the images of different cell lines.	49

List of Tables

4.1	Comparison of the proposed smart markers algorithm against different marker identification algorithms. The results are obtained on the training set.	35
4.2	Comparison of the proposed smart markers algorithm against different marker identification algorithms, in terms of the <i>precision</i> , <i>recall</i> and <i>F-score</i> measures. The results are obtained on the training set.	35
4.3	Comparison of the proposed smart markers algorithm against different marker identification algorithms. The results are obtained on the test set.	35
4.4	Comparison of the proposed smart markers algorithm against different marker identification algorithms in terms of the <i>precision</i> , <i>recall</i> and <i>F-score</i> measures. The results are obtained on the test set.	36
4.5	Cell-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the training set.	37
4.6	Cell-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the test set.	37

4.7	Area-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the training set.	37
4.8	Area-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the test set.	38

Chapter 1

Introduction

Automated imaging systems are becoming popular to analyze cellular events of fixed or live cells. These cellular imaging systems have potential not only for decreasing the processing time but also for reducing the human errors during the analysis. In almost all of the systems, cell segmentation constitutes the first step, which greatly affects the performance of the other system steps. Although there are several algorithms for the segmentation of fixed cell images from a light or a fluorescence microscope, there exist only few for the segmentation of live cells from phase contrast microscopy. In this paper, we focus on the implementation of a robust segmentation algorithm for live cells in culture media.

1.1 Motivation

In general, the previous studies have approached the cell segmentation problem in two different contexts: segmenting monolayer isolated cells and segmenting cells that grow in aggregates on layers. For the monolayer isolated cell segmentation, the studies first differentiate cell pixels from the background and then consider the connected components of the cell pixels as the segmented cells. To binarize the image, a threshold level can be obtained from the whole image [1], known as global thresholding, or more than one threshold level can be obtained from

different regions of the image separately [2–5], known as adaptive thresholding. Although adaptive thresholding is more preferable than global thresholding, it is still sensitive to noise and speckle. Another approach to obtain a binary image is to employ a clustering algorithm and then consider corresponding clusters as foreground (cells) [6].

Confluent cells need more complex techniques since thresholding and clustering approaches are not usually enough to separate these cells from each other. Segmentation using active-contour models is one of the well known solutions, that defines an energy function usually on the edge map of an image, associated with cell contours. Afterwards, it achieves segmentation by finding the contours that minimize the energy function [7–9]. Another solution is the watershed algorithm, that first defines markers, and then starts a flooding process from these markers. Traditional watershed algorithms consider the regional minima on the intensity/gradient map of an image as markers, reflecting the intensity differences between inside and outside of the cells [10–12], and/or on the distance transform of an initially segmented image, reflecting the shape characteristics of the cells [13–16].

For the segmentation of clumped cells, the previous studies mainly use marker-controlled watershed algorithms, because the traditional watershed algorithms are very sensitive to noise, causing over-segmentation. They first identify markers, each of which corresponds to a cell, and start the flooding process from these markers. The main difference between the marker-controlled and traditional watershed algorithms is that the marker-controlled watershed algorithm starts the flooding process from predefined markers whereas the traditional watershed algorithm starts the flooding process from regional minima. Applying iterative erosions [17] and modeling intensities/gradients by mixture of Gaussians [18] are some of the several methods for marker identification. The noteworthy point is that since each marker represents one cell, the definition of markers has an important role in the cell segmentation process. If there is a greater number of markers, the marker-controlled watersheds typically yield oversegmentation, in which there exist more segmented cells (markers) than the actual ones. Therefore, the studies commonly perform a merge process on the segmented cells afterwards [19–22].

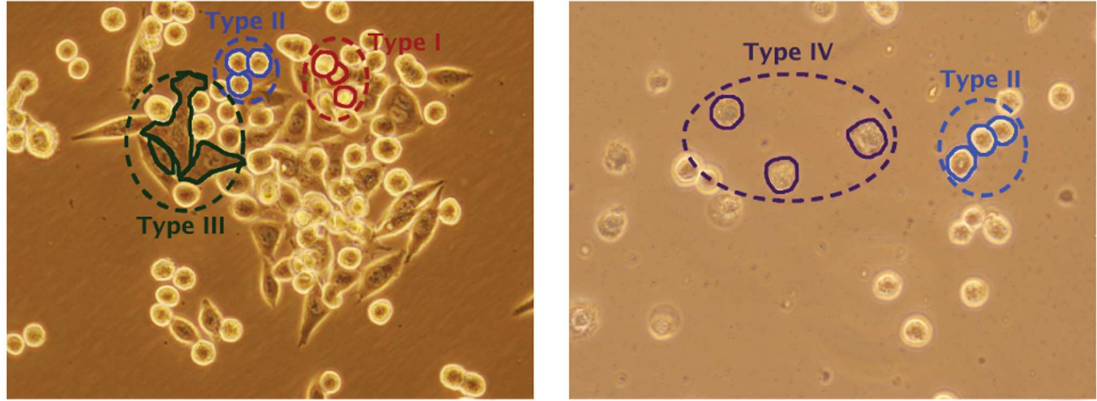


Figure 1.1: Example images of live KATO-3 gastric carcinoma cells. As shown in these images, the cells can be grouped into four morphological classes based on their visual characteristics. Examples from these groups are also indicated on the images.

Image segmentation in general is an ill-posed problem. The success highly depends on the intent of segmentation as well as the prior knowledge about the image content. This is especially the case for the problems, in which domain specific knowledge is necessary even for human subjects to achieve successful segmentations. Live cell segmentation is one of such problems. In live cell images, the cells of the same cell line or the same tissue may show different morphologies and intensity/texture characteristics. Moreover, these characteristics could be different from a cell line or a tissue to another. For example, KATO-3 gastric cancer cells can be grouped into four morphological classes based on their visual characteristics (Figure 1.1). The first group corresponds to round cells with relatively brighter inner and boundary pixels. The second one corresponds to round cells as well but these cells consist of relatively darker pixels in their centers and brighter pixels on their boundaries. The third group corresponds to non-circular cells that have relatively larger and irregular shapes and consists of high-gradient dark pixels. These cells also have brighter pixels on their boundaries. The last group corresponds to apoptotic cells whose inner regions and boundaries turn into matte and irregular. The algorithms with the capacity of incorporating this kind of biological knowledge into segmentation have a potential to improve the results. This will be our main motivation behind using domain specific knowledge, in the

form of visual characteristics of the cells, in our segmentation algorithm.

1.2 Contribution

In this thesis, we propose a new algorithm for the effective and robust segmentation of live cells. In the proposed algorithm, our main contribution is the incorporation of domain specific knowledge into the definition of a new set of “smart markers” for a watershed algorithm. In order to determine the smart markers, the proposed algorithm identifies different pixel groups with different visual properties, based on the biological background knowledge, and processes these groups with respect to each other, again using the background knowledge of different cell characteristics. Working with live cell images taken from the KATO-3 cell line, our experiments demonstrate that the proposed algorithm, which uses this new smart marker definition, is effective in finding better markers compared to its counterparts, which will in turn improve the segmentation performance of a marker-controlled watershed algorithm.

The proposed algorithm differs from the previous ones in two main aspects. First, it defines the smart markers based on the background knowledge specific to the image whereas the previous algorithms define them using intensity, gradient, and distance measures without considering the image specific properties. Second, the previous algorithms typically find more markers than the actual cells, resulting in oversegmentation, and hence, they usually necessitate using a merge process after their watershed algorithms. In contrary, the proposed algorithm can find more markers that are one-to-one mapped to the actual cells and can give less oversegmented results without using an external merge process.

1.3 Outline

The outline of the thesis as follows. In Chapter 2, we first define the segmentation problem and then give the existing approaches from literature. In Chapter 3, we

describe the proposed marker extraction and segmentation algorithm in detail. In Chapter 4, we give the details of the dataset, the procedure of parameter selection and evaluation. Afterwards, we give the experimental results of the proposed algorithm comparing to the other benchmark algorithms. Finally, in Chapter 5, we present conclusions and future research directions of our study.

Chapter 2

Background

In this chapter, we summarize automated cell segmentation algorithms in the literature. We can group automated cell segmentation algorithms into two main groups: group mainly focusing on segmentation of isolated cells and those also focusing on segmentation of touching (confluent) cells. While isolated cells can be segmented using simpler techniques, confluent cells need more effort because of their clustered structure. In the automated cell segmentation part, we describe commonly used algorithms for the segmentation of isolated cells, and then continue with describing algorithms for the segmentation of confluent cells.

2.1 Automated Cell Segmentation

To analyze biological systems at cellular level, imaging systems have been introduced recently [11, 13, 14, 23, 24]. These systems provide users to observe fixed and living cells in detail. Moreover, they can give quantitative results for a cell such as the probability of being a dead or living cell, which is very crucial in molecular and cellular biology research such as anti-cancer drug screening. Such analysis usually requires identifying the locations of cells (i.e., cell segmentation). On the other hand, manual segmentation of the cells may take too much time,

which makes the analysis impractical [25]. Furthermore, it is prone to human errors that have given rise to the subjectivity problem [18]. Therefore, integrating computer-aided systems into the segmentation process yields several advantages. Since it reduces the human interaction, it is capable of minimizing human errors and processing time. Thus, computers can be used to segment images with less errors and more efficiency provided that automated systems can be developed for the problem.

Cell segmentation problem can be considered in two contexts, segmentation of isolated and segmentation of confluent cells. As seen in Figure 2.1a, isolated cells are located on the plate alone; they are not touching each other. Gradient, intensity or color information can give sufficient information to segment such kind of cells. On the other hand, segmentation of clustered cells involves several challenges. Since cells are clumped together, they can create very complex visual appearances. An example of such clustered cells is shown in Figure 2.1b. They need more complex approaches than isolated cells need. In the remaining parts of this section, we give the details of methods commonly used for cell segmentation.

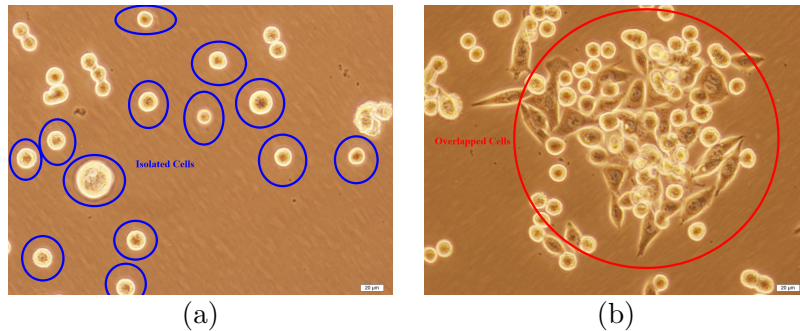


Figure 2.1: (a) isolated cells, (b) clumped cells

2.1.1 Thresholding

In many applications, thresholding, one of the simplest techniques, is employed to separate foreground pixels from background ones. This well-known approach, however, works successfully on the cells that have quite different characteristics from the background such as intensity [26], and gradient [1, 17]. The basic idea

behind thresholding is to define a value which can differentiate pixels of an image based on their intensity/gradient levels. A survey that explores thresholding methods and compares them can be found in [27].

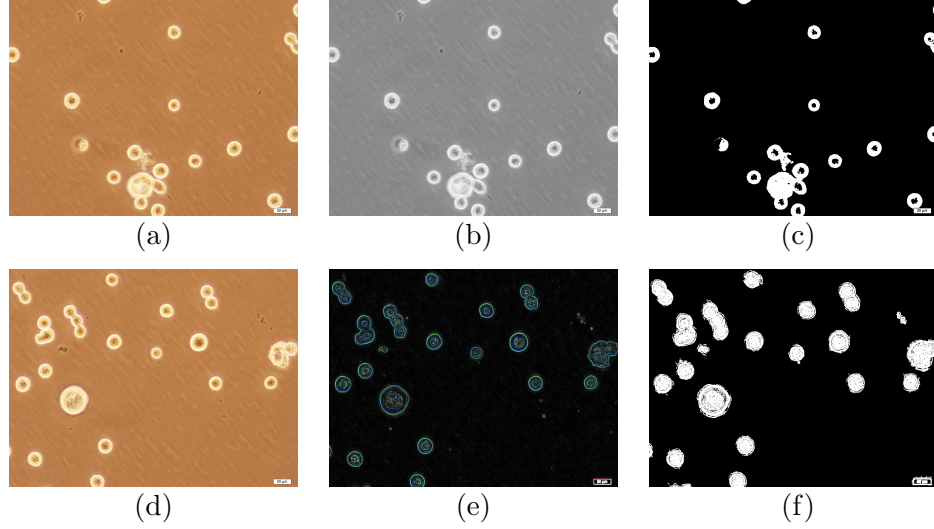


Figure 2.2: Illustration for thresholding on intensity and gradient values. (a) original KATO-3 image, (b) its gray-level image, (c) binary image after applying Otsu thresholding on the gray image, (d) another original KATO-3 image, (e) its gradient map obtained by Sobel operators, (f) binary image after applying Otsu thresholding on the gradient map.

Among the thresholding methods, one of the most referenced approaches is the Otsu’s method [28]. It clusters pixels in a way that the sum of within-class variances of each group is minimized. It is possible to apply the Otsu’s method to red, green and blue channels of images and then combine the results to obtain a binary image [12, 29]. Gradient [30] or gray-level images [11, 13] can also be employed as an input to the Otsu’s method. Figure 2.2 illustrates thresholding process on gradient and intensity images of the KATO-3 cell line, which is used as the primary cell line in our experiments. As seen in the figure, isolated cells can be extracted easily considering each connected component as one cell, but thresholding is not capable of segmenting touching cells, causing undersegmentation. Live cells appear on an image generally bright intensity values with less variation within. In this case, thresholding on gradient or intensity values will be enough to locate monolayer isolated cells. Illumination, however, can be a problem since the intensity values of cells and background pixels may vary. An alternative solution

is to obtain a threshold level for each pixel or each region, considering its local information [31–34], which is known as adaptive-thresholding. This solution may commonly overcome illumination related problems, but artifacts and clumped cells still cause problems that are the main challenges of live cell segmentation.

Clustering methods can also be employed for similar tasks since both methods, clustering and thresholding, group pixels such that they maximize inter-cluster distance and minimize intra-cluster distance. The difference is that thresholding approaches define a threshold value to specify clusters whereas clustering approaches define cluster representatives to perform the same task. After we employ a clustering algorithm, we can use corresponding clusters to define cells or cell boundaries. As illustrated in Figure 2.3, the yellow labeled cluster represents boundary regions of cells, which can be used to segment isolated cells. However, we need to incorporate other techniques for the segmentation of confluent cells. This thesis defines groups of pixels and uses them for the purpose of confluent cell segmentation. Depending on images and cell lines used, various features such as intensity [11, 23], gradient [30], red, green, and blue color [35] can be used for clustering.

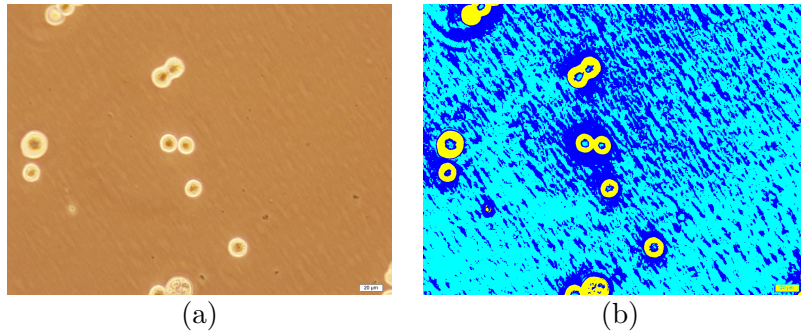


Figure 2.3: (a) Original KATO-3 image, and (b) labeled image after applying k-means clustering on red, green, and blue values of pixels.

2.1.2 Active Contour Models

Active contour models are first introduced in [36]. These models are also known as *snakes* that aim to find salient image contours. Their main advantage is that

they can fit closed, continuous, and smooth boundaries automatically. However, the algorithm needs initial points to locate a curve and these initial points greatly affect the results. After initialization, it defines an energy function using the curve and then minimizes the function by evolving the curve iteratively. The function employs external and internal forces while seeking the optimal contour.

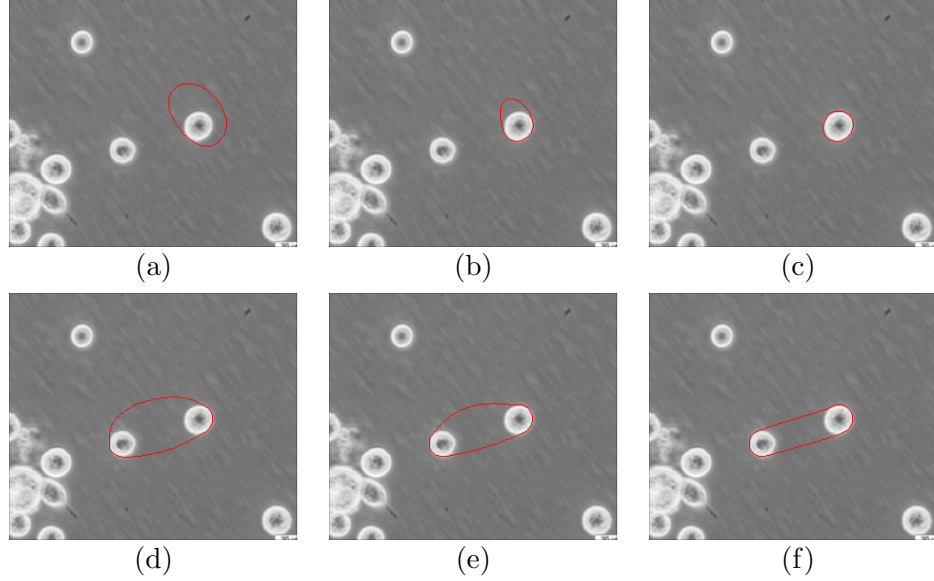


Figure 2.4: Segmentation results obtained using active contour models on an example live cell image. (a)-(c) using a smaller initial curve: after (a) 20 iterations, (b) 80 iterations, and (c) 200 iterations. (d)-(f) using a larger initial curve: after (d) 10 iterations, (e) 50 iterations, and (f) 500 iterations.

The forces evolve the curve toward pixels with high gradient values. In this case, the algorithm locates its optimal contour nearby the high gradient regions. Considering this situation, traditional active contours work well on the boundary with high gradient values. Figure 2.4 depicts how an active contour model performs on an isolated live cell. One observation from this figure is that the intensity difference between background and cells is quite distinctive which means that the boundary pixels of cells have relatively higher values. Snakes fit a close curve on an object if gradient is enough to represent its boundary. On the other hand, it may fail to locate a curve on an object having smooth boundary since the object has pixels with lower gradient values on its boundary. This drawback is discussed in [37]. Another drawback is that active contour models largely depend on the initial curve and parameters. As seen in the last row of Figure 2.4, when

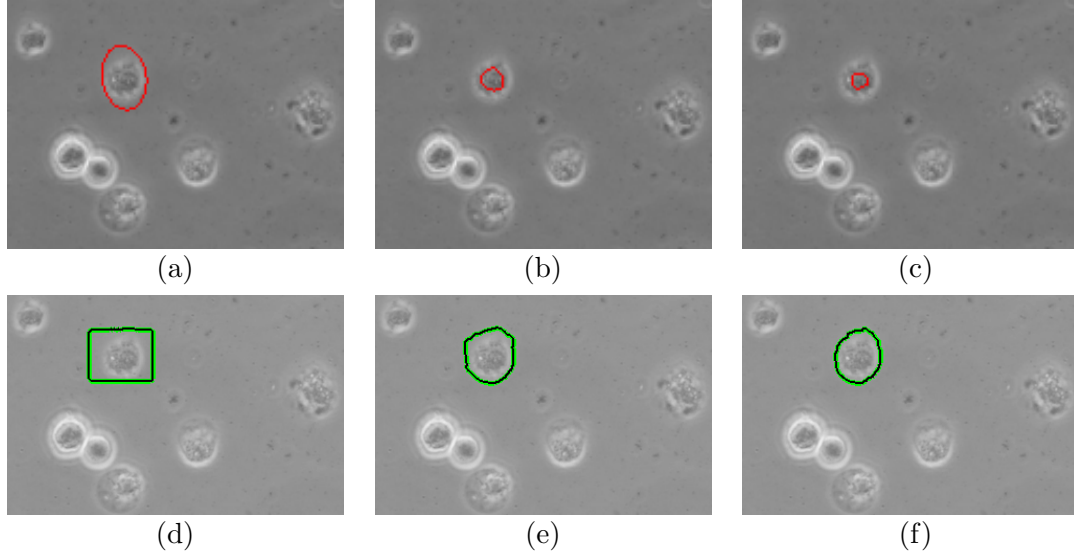


Figure 2.5: Illustration of snakes and region-based active contours on an isolated cell having boundary with lower gradients. (a)-(c) correspond to the snake algorithm, and (d)-(f) correspond to the region-based active contour model.

we employ a larger initial curve with the same parameters, the active contour cannot fit well even after 500 iterations which causes undersegmentation. This drawback is attempted to be overcome in [38] by estimating the initial position of the curve using morphological operators; however this approach is not capable of handling the segmentation of clumped cells. The snake algorithm is improved in [37] such that the algorithm does not depend on gradient values along the contour. This algorithm, known as region-based active contours, considers gradient values inside and outside of the contour. This makes region-based active contour models more robust for finding smooth boundaries. Figure 2.5 shows how snakes and active contour models work on boundaries with lower gradients. The region-based active contour model fits the curve very close to actual cell boundaries while the snake algorithm cannot detect actual cell edges, and instead it locates a very small curve inside the cell. Moreover, it can find optimal boundaries regardless of the initial curve location on the image. A sample image is illustrated on Figure 2.6. Although the algorithm can find more accurate boundaries than snake can find, it is not able to separate confluent cells from each other that generally causes undersegmentation.

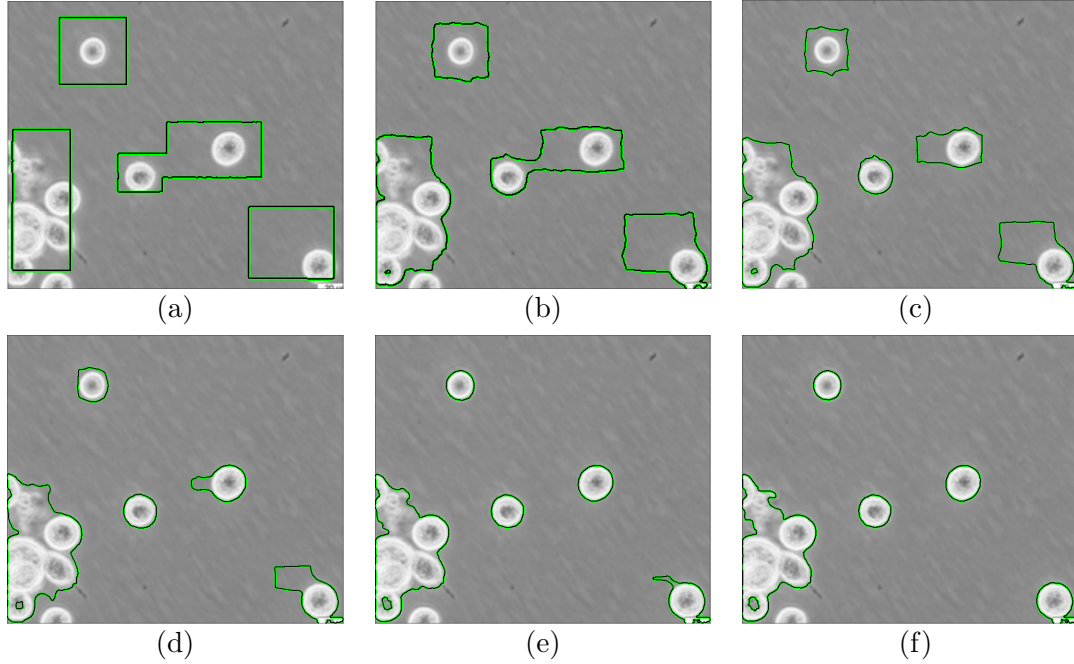


Figure 2.6: Results using region-based active contour models on an example live cell image: after (a) 1 iteration, (b) 100 iterations, (c) 200 iterations, (d) 300 iterations, (e) 400 iterations, and (d) 500 iterations.

2.1.3 Watersheds

The watershed algorithm is a commonly used approach to overcome confluent cell segmentation problem [39]. Considering the domain, several features such as color, intensity, and gradient can be used in watershed algorithm. It first defines seeds and grows starting from them. In other words, if we consider seeds as the minima of catchment basins, water starts flooding from these minima. Pixels, in which two floods merge, are defined as ridges. These ridges can be used to segment cells. Figure 2.7 shows a topological surface of a sample KATO-3 image according to the intensity. The algorithm can segment these cells successfully since each marker corresponds to one actual cell.

Traditional watershed algorithms define seeds based on intensities or a distance map, on which each pixel has the minimum distance value to the background. Local minima of intensity or inverse of the distance map are considered as seeds. This approach, however, is very sensitive to noise. Artifacts can cause

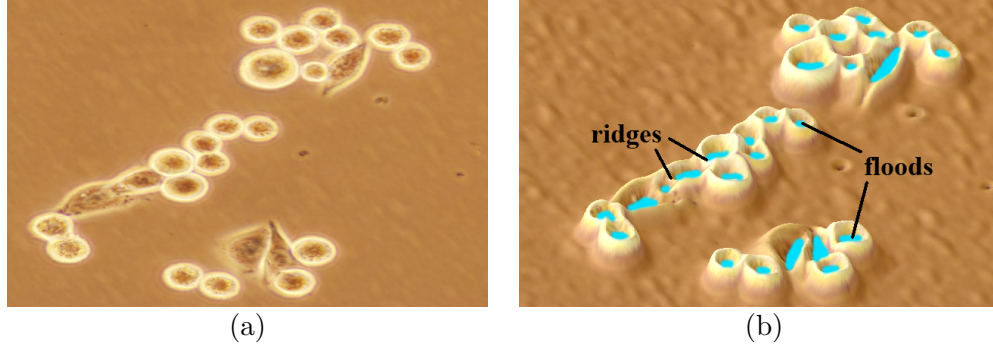


Figure 2.7: Illustration of a watershed algorithm. (a) Original image, and (b) flooding process starting from the darker regions.

the definition of extra seeds, which generally yields oversegmentation or false detections. One solution is to remove relatively small minima by applying h-minima transform. The transform considers regional minima whose depth is smaller than an h-value as noise and suppress all such minima. The critical point is how the h-value is defined. Selecting larger h-values removes lots of noise whereas this also removes small cells. On the other hand, selecting smaller h-values does not remove all the noise. It is possible to select the h-value adaptively according to size of objects [15,16]. Live cells, however, may have different sizes because they are not fixed on the plate. In this way, it is not suitable to use the same h-value for all cells in an image. Another solution is to merge oversegmented regions. The merging criteria should be addressed quite well in order to get accurate results. Both size and compactness of objects can be taken into the consideration for the merging criteria [1, 22]. Regions are merged based on the number of pixels on the boundary between objects and their area size [19]. In [20], they add a new constraint that is the strength value of the boundary to [19]. Strength value, defined on a gradient map, is the mean value of all pixels along the border.

Like h-minima transform, these approaches mainly depend on the parameters. It is not possible to tune parameters for every image. Instead of applying these post- and pre-processing techniques, marker-controlled (seeded) watersheds have been proposed to segment the confluent cells. These watershed algorithms employ predefined markers as seed points and starts water to rise from these seeds. Defining markers, which is the main focus of the thesis, is very crucial

since each marker should correspond to one cell after the segmentation. If we define more markers than actually needed, it will lead to oversegmentations or false detections. Defining less number of markers yields undersegmentations or misses. In [1, 13], shape information is integrated into the definition of markers using an inner distance map. The shape information is also incorporated in [2, 17, 40] using morphological operators. To define markers, these studies need a structuring element, specific to the cell shape. However, using a single structuring element may cause problems for the segmentation of cells, because there can exist a variety of cell shapes in a live cell image, as seen in Figure 2.1. In [14], shape and intensity information is integrated into a seeded watershed algorithm. They obtain markers from the combination of intensities and the inner distance map.

2.1.4 Other Methods

Several algorithms segment confluent cells making use of prior shape information. In [26, 35], ellipses are located considering that cells have round-like shape. In [35, 41] cells are found separating segmented region from their concave points. Here, they obtain the concave points, and separate the region locating a line between these points. In [18], cells are modeled with Gaussian mixtures and cell segmentation is formulated as a cluster analysis problem. As an alternative, it is proposed to use graph-based segmentation techniques, which consider the cell segmentation as a graph-cut problem [42]. Although all of these algorithms employ prior cell knowledge in the segmentation procedure, they do not consider the case that cells can have different visual characteristics. One reason might be that the majority of these algorithms are developed on cells of fluorescence images, which are fully attached to the plate causing the cells have a similar appearance. However, live cells are not fully attached to the plate, which causes them to have different visual characteristics.

Chapter 3

Methodology

3.1 Smart Markers Algorithm

The proposed algorithm relies on defining three basic types on the cell pixels—according to the intensities and gradients of these pixels and their surroundings, associating these basic types with cells of different characteristics, and extracting the markers on each of these basic types by considering the morphological characteristics of their associated cells. The details of this algorithm are explained in the following subsections. A schematic overview of the algorithm is provided in Figure 3.1.

In this work, we develop our algorithm focusing on the KATO-3 human gastric cancer cell line. We consider the characteristics of its cells in the definition of the basic pixel types and the markers. Nevertheless, this idea can also be applied to other cell lines or tissues, provided that the basic types reflecting the characteristics of their cells are defined. In our experiments, we also obtain preliminary results on four different cell lines to explore the applicability of this algorithm to others.

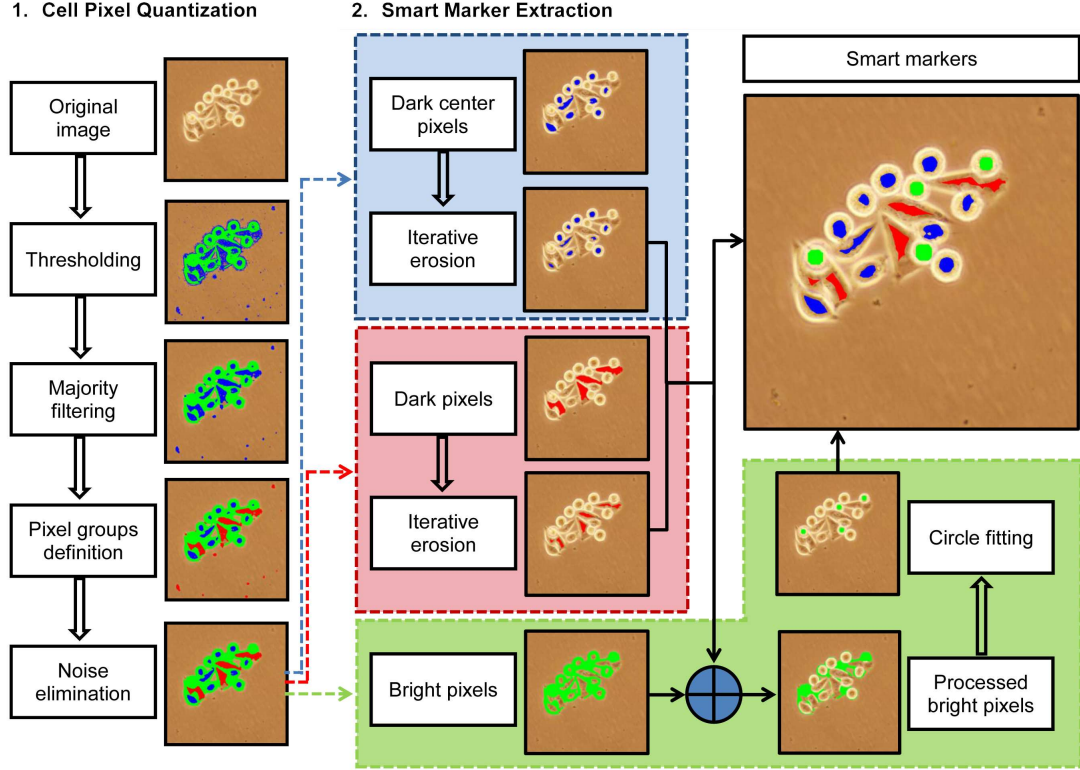


Figure 3.1: Schematic overview of the proposed algorithm.

3.1.1 Cell Pixel Quantization

This part consists of transforming an image into three basic types of pixel groups, each of which corresponds to a cell region of different characteristics. These types correspond to (i) the bright pixels, (ii) the dark pixels fully surrounded by bright pixels, and (iii) the dark pixels only partially surrounded by bright pixels. They are herein referred to as *bright*, *dark-center*, and *dark* pixels, respectively. These three pixel types are used for characterizing the four morphological classes¹ of the KATO-3 gastric cancer cells.

¹As explained in the introduction and illustrated in Figure 1.1, the KATO-3 gastric cancer cells can be grouped into four classes: *Type I* cells are round and consist of bright pixels both in their centers and on their boundaries. *Type II* cells are also round but contain dark pixels in their centers and bright pixels on their boundaries. *Type III* cells have irregular shapes and contain the dark pixels usually partially surrounded by the bright pixels. *Type IV* cells are apoptotic and contain mostly dark pixels but barely bright pixels.

Particularly, we employ the bright pixels for characterizing Type I cells as well as the boundaries of the others, the dark-center pixels for Type II cells, and the dark pixels for both Type III and Type IV cells. Cell pixel quantization starts with identifying bright and dark cell pixels. The *bright* pixels correspond to high intensity regions in the image. Hence, we obtain them by thresholding the gray-level image with Otsu's method [28], which automatically computes the threshold t_{gray} on the intensity values. The *dark* pixels correspond to relatively darker regions with high gradient values. Here one should note that the dark cell regions have a similar intensity distribution with that of the background. Therefore, using only the intensities, without considering the gradient values, would yield errors in pixel quantization. In this work, we use the Sobel operators on the gray-level intensities to define the gradient values. Computing a new Otsu threshold t_{sobel} on these gradients, the dark pixels are defined as the pixels whose gray-level intensities are less than t_{gray} and whose gradients are greater than $k \cdot t_{sobel}$. Here, the Sobel threshold is multiplied by a constant k since our experiments reveal that relatively lower gradients should also be considered in the dark pixel definition. Let I_{ij} and S_{ij} be an intensity and gradient value of the gray level pixel, $P(i, j)$, respectively and Q be the quantized resulting image. The bright, dark, and background pixels are defined as follows

$$Q(i, j) = \begin{cases} \text{bright} & \text{if } P_{ij_{gray}} \geq t_{gray} \\ \text{dark} & \text{else if } P_{ij_{grad}} > k \cdot t_{sobel} \\ \text{background} & \text{otherwise.} \end{cases}$$

For live cell images given in Figure 3.3, the quantized images are illustrated in Figure 3.3b. After this quantization, the dark pixels are further grouped into two based on whether they are fully surrounded by the bright pixels; that is, some of the dark pixels are identified as the dark-center pixels. However, there usually exists noise in the quantized pixels, which leads to errors in the definition of the dark-center pixels. Thus, we postprocess the quantized pixels to alleviate the noise. For that, we first eliminate the narrow dark pixel regions around the boundaries of the bright regions, explained in Algorithm 1 and then apply a majority filter on the quantized pixels. The majority filter is an image processing filter that provides a very appropriate way to smooth an image (Figure 3.2).

The filter replaces each pixel in the image with the majority value among its neighboring pixels. The neighborhood is defined by using a structuring element.

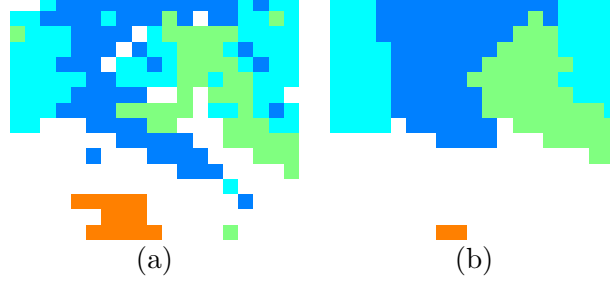


Figure 3.2: Illustration of the majority filtering with a square structuring element (size = 4): (a) before and (b) after.

Input: Quantized image Q , length of a line $length^2$

Output: Modified quantized image Q

$C_{bright} \leftarrow$ 8-connected components of *bright* pixels in Q

for all connected component $C_k \in C_{bright}$ **do**

for all boundary pixel $P_{(i,j)} \in C_k$ **do**

$S_{horizontal} \leftarrow \{P_{(i,j)}, P_{(i+1,j)}, \dots, P_{(i+length,j)}\}$

$n_{dark} \leftarrow$ number of *dark* pixels in $S_{horizontal}$

if $n_{dark} < length/2$ **then**

 assign all pixels in $S_{horizontal}$ as *background*

end if

$S_{vertical} \leftarrow \{P_{(i,j)}, P_{(i,j+1)}, \dots, P_{(i,j+length)}\}$

$n_{dark} \leftarrow$ number of *dark* pixels in $S_{vertical}$

if $n_{dark} < length/2$ **then**

 assign all pixels in $S_{vertical}$ as *background*

end if

end for

end for

Algorithm 1: Eliminating narrow *dark* pixels around *bright* regions.

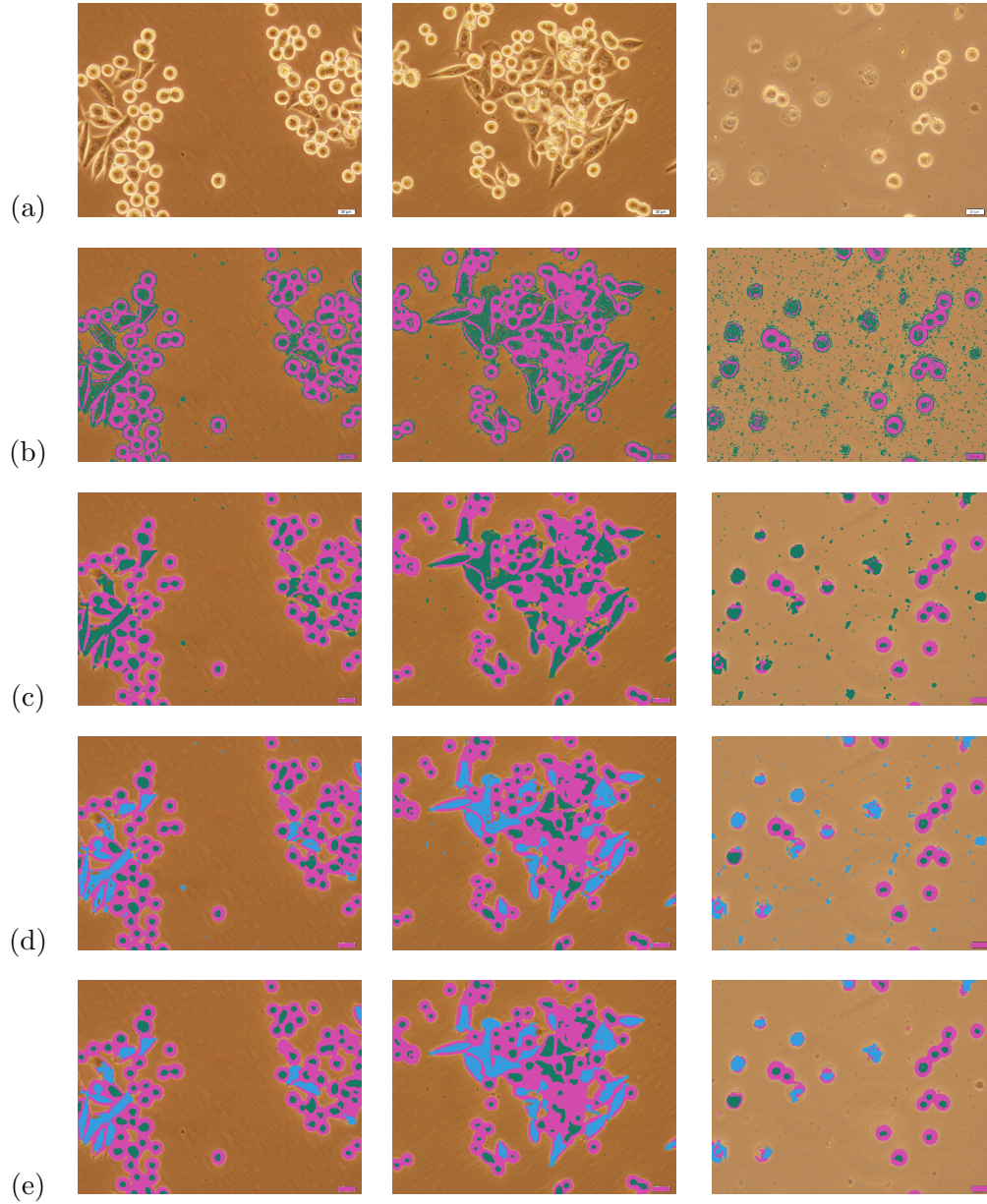


Figure 3.3: Illustration of the cell pixel quantization step on three exemplary images. (a) start with an original image, (b) identify *bright* and *dark* pixels using the intensity and gradient information, (c) remove narrow dark pixels around of bright pixels and then apply majority filtering, (d) identify some of the dark pixels as *dark-center* pixels, (e) eliminate noise, artifacts and small background regions. In this illustration, the bright, dark, and dark-center pixels are shown with pink, green, and blue, respectively.

For the example live cell images shown in Figure 3.3a, the quantized pixels obtained by this process are illustrated in Figure 3.3c. In this figure, the bright and dark pixels are shown in purple and green, respectively. After this noise elimination, we identify the *dark-center* pixel groups as follows: We consider all pixels except the bright ones and find the connected components on these pixels. Let C_i be the i th connected component and d_i and b_i be the numbers of the dark and background pixels in the component C_i , respectively. The dark pixels in C_i are identified as the dark-center pixels if $d_i > b_i$. Otherwise, they remain as the dark pixels (Algorithm 2). Figure 3.3d illustrates the quantized pixels obtained at the end of this step. Here the dark-center pixels are shown in blue.

Input: Quantized image Q

Output: Modified quantized image Q

```

 $C_{darkbg} \leftarrow$  8-connected components of dark and background pixels in  $Q$ 
for all connected component  $C_k \in C_{darkbg}$  do
     $n_{dark} \leftarrow$  number of dark pixels in  $C_k$ 
    if  $n_{dark} > size(C_k)/2$  then
        assign all dark pixels in  $C_k$  as dark-center
    end if
end for

```

Algorithm 2: Defining *dark-center* pixels.

The final step is to eliminate holes and artifacts from the pixel groups. First, we fill the holes in between the cell pixels provided that the holes are smaller than an area threshold T_{area} . In our experiments, we observe that the main source of noise and artifacts is the dark components. They may correspond to small noisy regions as well as relatively larger artifacts usually found in the background (see the second column of Figure 3.3). These larger artifacts typically do not contain any bright pixels on their boundaries. Thus, following these observations, we define two post-processing steps: First, we eliminate the dark components if they are smaller than the area threshold T_{area} . Second, we eliminate the dark components that do not contain any bright pixels on their boundaries. This algorithm is summarized in Algorithm 3 and Algorithm 4. The quantized pixels obtained at the end of these elimination procedures are shown in Figure 3.3e. We use these quantized pixels to define our smart markers.

Input: Quantized image Q , area threshold T_{area}
Output: Modified quantized image Q

$C_{bg} \leftarrow$ 8-connected components of *background* pixels in Q
for all connected component $C_k \in C_{bg}$ **do**
 $n \leftarrow$ number of pixels in C_k
 if $n < T_{area}$ **then**
 if C_k has *dark-center* neighbors **then**
 assign C_k as *dark-center*
 else if C_k has *dark* neighbors **then**
 assign C_k as *dark*
 else
 assign C_k as *bright*
 end if
 end if
end for

Algorithm 3: Eliminating small background regions.

Input: Quantized image Q , area threshold T_{area}
Output: Modified quantized image Q

$C_{dark} \leftarrow$ 8-connected components of *dark* pixels in Q
for all connected component $C_k \in C_{dark}$ **do**
 $n \leftarrow$ number of pixels in C_k
 if $n < T_{area}$ **then**
 assign C_k as *background*
 else
 $n_{bright} \leftarrow$ number of *bright* pixels on the boundary of C_k
 if $n_{bright} == 0$ **then**
 assign C_k as *background*
 end if
 end if
end for

Algorithm 4: Eliminating artifacts.

3.1.2 Smart Marker Extraction

The proposed algorithm defines the markers for each of the three pixel types separately, according to the characteristics of the regions that each type corresponds to. Since the markers are defined considering the background knowledge of the corresponding region characteristics, it is expected to find more markers that are one-to-one mapped to the actual cells, and thus, to obtain less under and oversegmented results.

In order to define the markers on the dark and dark-center pixels, we employ an iterative erosion algorithm. This algorithm erodes the given pixel groups iteratively until the size of a group falls below a threshold. In our work, we select this threshold separately for the dark and dark center pixels, considering their region characteristics. Since the dark-center pixels usually correspond to relatively smaller regions compared to the dark pixels, we use a size threshold T_{size} for the dark pixels and the half of it ($T_{size}/2$) for the dark-center pixels. Similarly, we use a disk structuring element with a radius of r_{disk} for the erosion of the dark pixels and its half ($r_{disk}/2$) for that of the dark-center pixels. The iterative erosion algorithm on the dark and dark-center pixels is illustrated in Figures 3.4 and 3.5, respectively.

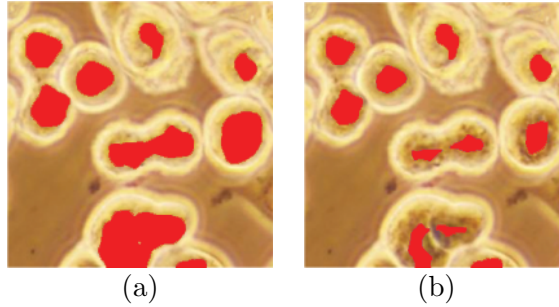


Figure 3.4: Illustration of the iterative erosion algorithm on the *dark center* pixels: (a) before and (b) after.

To define the markers on the bright pixels, we take the following observation into consideration. The bright pixels can be found both inside a particular class of cells and the boundaries of the others. To alleviate the negative effects of the boundaries, we first dilate the previously found markers and then locate

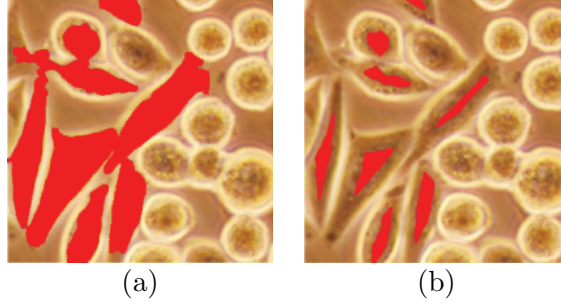


Figure 3.5: Illustration of the iterative erosion algorithm on the *dark* pixels: (a) before and (b) after.

circles on the remaining bright pixels using the modified version of the circle-fit algorithm [43]. As seen in Figure 3.6, the circle fitting algorithm locates circles not only inside a particular class of cells but also on the boundaries of the others, which leads to oversegmentation and false detection unless we exclude the boundary pixels.

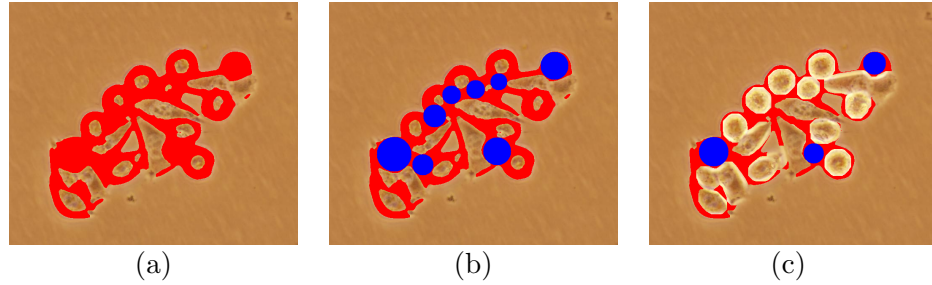


Figure 3.6: Illustration of effects of the boundaries: (a) *bright* pixels, (b) circle fit algorithm on all *bright* pixels and (c) circle fit algorithm on the remaining *bright* pixels after we exclude the dilated version of the previously found markers.

Therefore, we first dilate the pre-found markers and then exclude them from the bright pixels. On the remaining bright pixels, the circle fitting algorithm is run. In this algorithm, we iteratively locate circles on the given pixels, starting from the largest one, provided that the size of a circle is larger than the threshold T_{size} . Since the algorithm locates the largest possible circles on a binary image, it generally yields undersegmentation for confluent cells. By employing an additional constraint, that is the circle boundaries should be close enough to the non-bright pixels, the problem can be alleviated. Here, we define two internal parameters. One of them is the distance threshold value that is the maximum

distance value a boundary pixel can have to be considered as close to the non-bright pixels. We select the threshold value as 15, considering the resolution and magnification of images in our dataset. Another parameter is the number of boundary pixels of a circle that are close to the non-bright pixels. Considering this constraint, the algorithm does not locate a circle unless this number is larger than the number of half of the boundary pixels. Figure 3.7 shows effects of the constraint on the algorithm. The threshold value used in this constraint is the internal parameter of the system and is selected experimentally. It is not considered in the parameter selection step.



Figure 3.7: Illustration of the effect of the constraint: (a) *bright* pixels, (b) circle fit algorithm without employing the constraint and (c) circle fit algorithm with employing the constraint.

The final result of the circle-fit algorithm on the bright pixels is illustrated in Figure 3.6(c). Located circles are considered as markers, especially representing the Type-I cells. Together with dark and dark-center markers, they are employed for the marker controlled watershed algorithm in order to segment four different types of cells.

The aim of this work is to find more accurate markers to be used in a marker-controlled watershed algorithm. We also explore the effect of the use of these markers in a marker-controlled watershed algorithm. Therefore, we also implement the flooding process. Our experiments showed that the smart marker definition increases the success in terms of marker localization. This, in turn, is expected to increase the success of a watershed algorithm. To examine this, we implement a watershed algorithm that takes the smart markers as starting locations and grows them by making use of the marker types and the pixel groups

(*dark*, *dark-center*, and *bright* pixels). Here we use the geodesic distance from a pixel to a marker boundary as the growing criterion. Let M_{dark} , M_{center} , and M_{bright} be a set of smart markers defined on dark, dark-center, and bright pixels, respectively. In the watershed algorithm, we first grow the markers M_{dark} on dark pixels as long as the Euclidean and geodesic distances from a dark pixel to the corresponding marker boundary are equal to each other. This equality constraint is defined to prevent flooding into dark pixels that belong to missing cells with unidentified markers. Then, we repeat the same procedure to grow the markers M_{center} , this time on dark-center pixels. Finally, we combine the grown markers with the centroid of the markers M_{bright} and grow all of them on bright pixels. Here, we identify the most distant pixels that each marker can grow into. For that, for each marker M_i , we find the first bright pixel p_i that is adjacent to the background and that M_i grows into and define the maximum distance as the geodesic distance from p_i to the closest boundary of M_i plus an offset value, which is set to 10 in the experiments. This distance constraint is defined to prevent flooding into pixels of missing cells with unidentified markers as well as background pixels that are incorrectly assigned to the bright pixel group. At the end, we postprocess the results by applying the majority filter on the grown areas and filling the holes in each segmented cell. Note that in the experiments, we will compare our marker definition algorithm with others. For these comparison algorithms, we will grow their markers on their initial masks by considering the same distance constraint and applying the same postprocessing. Figure 3.8 depicts the resultant boundaries with their corresponding markers.

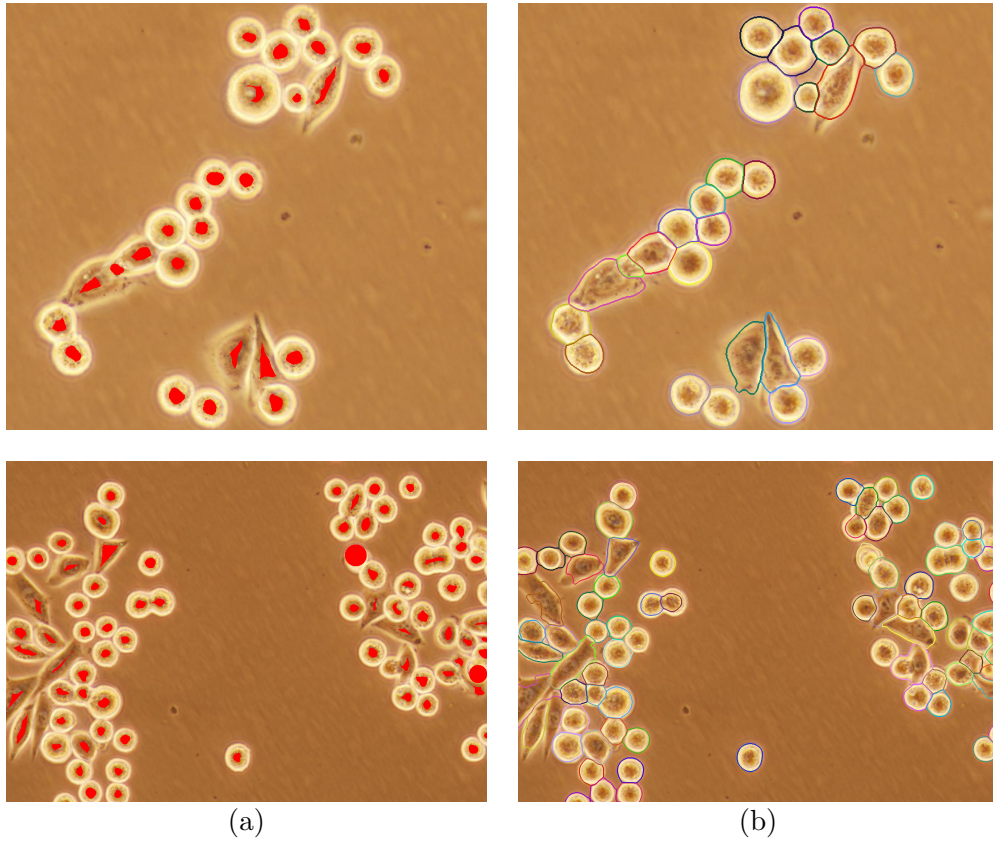


Figure 3.8: For two example images, (a) *smart* markers identified by the proposed algorithm, and (b) cell boundaries obtained after growing these smart markers by employing dark, dark-center, and bright pixels.

Chapter 4

Experimental Results

4.1 Dataset

Five different cell lines are used in the experiments. The human liver cancer cell line (HUH-7) and the human breast cancer cell line (MCF-7) were inoculated in complete growth medium composed of DMEM, with 10% FBS, 1% NEAA and 1% Penicilin/Streptomycin. The human endometrial carcinoma cell line (MFE-296) was cultivated in growth medium containing 40% RPMI 1640, 40% MEM (with Earle salts), 10% FBS, 2mM L-glutamine and $1 \times$ insulin-transferrin-sodium selenite.

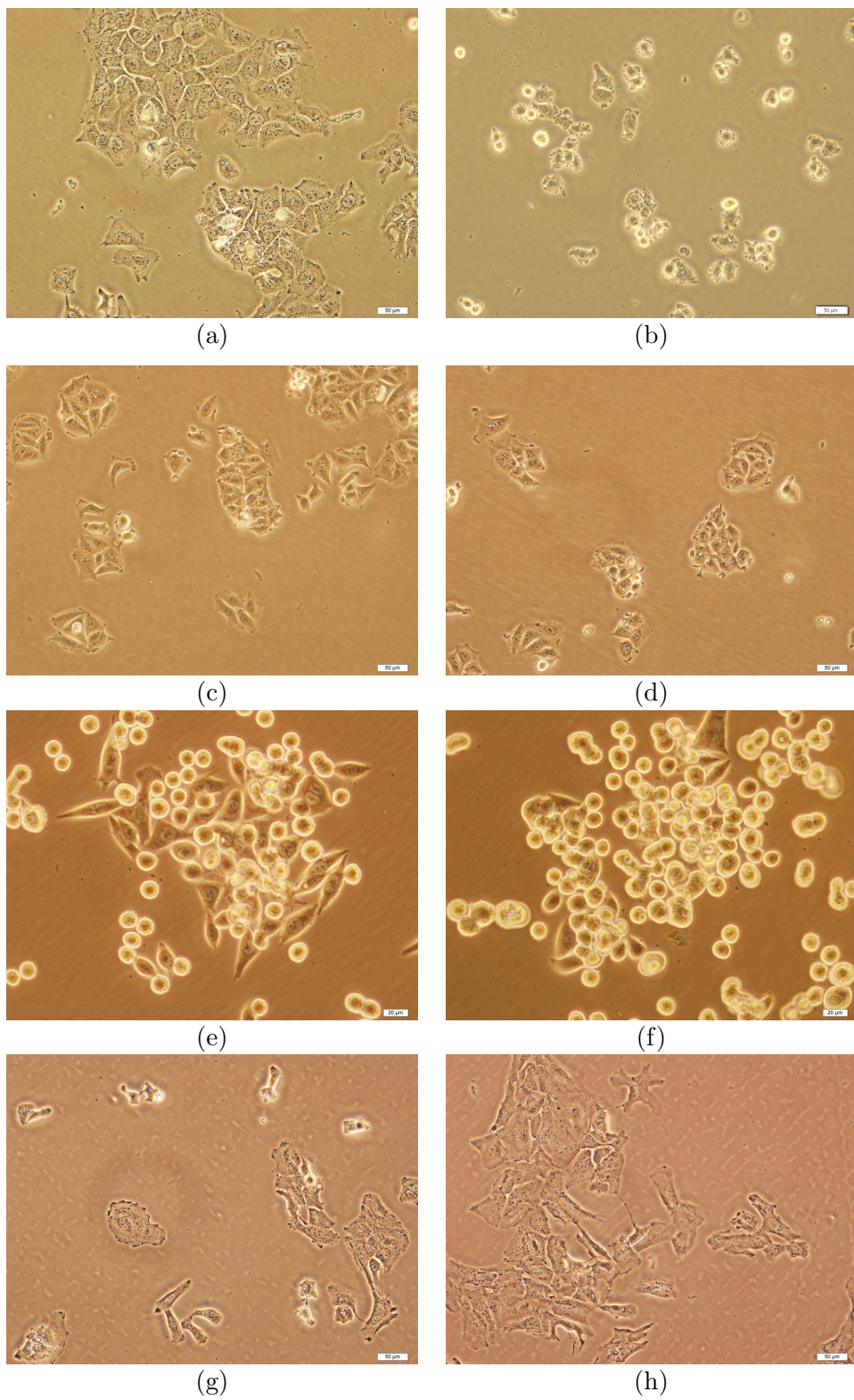


Figure 4.1: Example images obtained from different cell lines: (a)-(b) MFE-296, (c)-(d) MCF-7, (e)-(f) KATO-3, (g)-(h) HUH-7.

The human gastric cancer cell line (KATO-3) was inoculated in growth medium containing High glucose (4500 mg/L D-Glucose) DMEM with 10% FBS, 1% NEAA, 1% Penicilin/Streptomycin, and 1% L-glutamine. The human breast cancer cell line (SK-BR-3) was inoculated in complete growth medium composed of HyClone MCCOY'S 5A, together with 10% FBS and 1% L-glutamine. All cell lines were incubated in 37 °C, 5% CO₂, 95% air containing incubators. Figure 4.1 shows exemplary images obtained from different cell lines. As shown in this figure, cells can have different visual characteristics since they are not fully attached to the plate. For instance, in Figure 4.1(a-d) cells have different shapes. Some of them are small and have circular shape while others are relatively large and have irregular shape. In Figure 4.1(c-d) nucleolus of most of the cells, small and dark regions in a cell, are visible while others are not visible. Moreover, they have bright pixels on their boundaries. To sum up, although having different visual characteristics make the segmentation problem much more difficult, incorporating such knowledge into the algorithm can facilitate it.

We conduct our main experiments on 44 live cell images of the KATO-3 human gastric cancer cell line. The dataset contains a total of 1954 cells most of which grow in clumps on layers. Each image has a resolution of 1360×1024 pixels. The images are captured by a digital (Olympus DP72, Tokyo, Japan) microscope with a 40 $\times$ objective lens. The cells are annotated by our biologist collaborators, manually drawing their boundaries. We will use the centroids of these annotated cells for marker-based evaluation and their boundaries for area-based evaluation. In our experiments, the images are randomly divided into training and test sets. The training set includes 474 cells of 10 different images whereas the test set includes 1480 cells of the remaining 34 images. The cells in the training set are used to estimate the parameters of our algorithm as well as those that we use in our comparisons. The cells in the test set are not used in the parameter estimation at all ¹.

Marker-controlled watershed algorithms first identify the markers on an image

¹If the parameters were selected and the results were reported on the same image set, there would be a risk of obtaining optimistic results.

and then start the flooding process from these markers. The success of the segmentation is closely related with how well the markers are identified on the image. One can obtain better segmentation results if there is one-to-one correspondence between the markers and the actual cells. Since the correct identification of the markers greatly affects the segmentation results as well as the main contribution of this paper is on the marker definition, we first report the experimental results in terms of the markers, but not the segmentation boundaries. However, it is also possible to apply a watershed algorithm on the markers to obtain the boundaries. In the previous chapter, we discuss how these markers can be grown in a watershed algorithm by using dark, dark-center, and bright pixel information. The results of such a growing algorithm are also given. Please note that it is also possible to grow these markers by defining other algorithm. This is one of the future work of this thesis.

In our experiments, we evaluate the results both visually and quantitatively. For that, we consider three different evaluation criteria: marker-based, evaluation for assessing how well markers are located and cell-based and area-based evaluations for assessing how well markers are grown and boundaries are delineated. For marker-based evaluation, we define the centroid markings of the cells as the gold standards and the centroids of the identified markers as the computed cells and use a distance-based evaluation algorithm to obtain the quantitative results. In this evaluation algorithm, each marker (computed cell) is matched to every gold standard cell provided that the distance between the marker and the gold standard cell is less than a predefined distance threshold. By making use of these matchings, we define one-to-one match, oversegmentation, undersegmentation, false detection, and miss as follows:

- A marker (or a gold standard cell) corresponds to *one-to-one match* if the marker is matched to a single gold standard cell that is not matched with any other markers.
- A gold standard cell corresponds to *oversegmentation*, if more than one marker is matched to this gold standard cell. The number of such markers is considered in reporting the quantitative results.

- A marker corresponds to *undersegmentation* if it is matched more than one gold standard cell. The number of such gold standard cells is considered in reporting the quantitative results.
- A marker corresponds to *false detection*, if it is not matched to any gold standard cells.
- A gold standard cell corresponds to *miss*, if none of the markers are matched to this gold standard cell.

Cell-based and area-based evaluations can be done after finding the boundaries of cells (i.e., after the growing process). Here cell-based evaluation tells us how much percentage of the grown regions (markers) one-to-one correspond to the gold standard regions (cells). On the other hand, area-based evaluation tells us how well the boundaries of cells are delineated if these cells correspond to one-to-one matches.

For cell-based evaluation, a segmented cell S is said to be true if at least half of its pixels overlap a gold standard cell G and at least half of the pixels of G overlap S . That is, the pixels of a segmented cell are not considered as true positive if there is no one-to-one correspondence between this cell and a gold standard cell.

For both marker-based and cell-based evaluations, we use the number of one-to-one matches and compute precision, recall, and F-score. The definition of precision and recall are given as follows.

$$precision = \frac{\text{number of one to one matches}}{\text{number of computed cells}} \quad (4.1)$$

$$recall = \frac{\text{number of one to one matches}}{\text{number of actual cells}} \quad (4.2)$$

In area-based evaluation, we also use the same measures, but this time, they are defined on pixels. For that, we first find the true segmented cells after growing the markers, as in the case of cell-based evaluation, and then calculate precision,

recall, and F-score by considering the true positive pixels of these true segmented cells. In this case, the precision and recall measures are defined as follows.

$$precision = \frac{\text{number of true positive pixels}}{\text{number of pixels of computed cells}} \quad (4.3)$$

$$recall = \frac{\text{number of true positive pixels}}{\text{number of pixels of actual cells}} \quad (4.4)$$

For all evaluations, we calculate the F-score, which is defined as the harmonic mean of precision and recall, as

$$F\text{-score} = \frac{2 \cdot precision \cdot recall}{precision + recall} \quad (4.5)$$

4.2 Parameter Selection

The proposed algorithm has five external model parameters. The first three of these parameters are used for cell pixel quantization whereas the other two are used for smart marker extraction. These parameters are the Sobel threshold constant k , the size W of the majority filter, the area threshold T_{area} , the size threshold T_{size} , and the radius r_{disk} of the structuring element. In our experiments, we consider all possible combinations of the following parameter sets $k = \{0.2, 0.3, \dots, 0.6\}$, $W = \{9, 11, \dots, 21\}$, $T_{area} = \{750, 1000, \dots, 1500\}$, $T_{size} = \{750, 1000, \dots, 1500\}$, and $r_{disk} = \{5, 6, \dots, 11\}$. From all possible combinations of the parameter sets, we select the one that gives the maximum F-score on the training cells. This selection automatically evaluates the combinations based on their F-scores and does not involve any manual or visual examination. After this procedure, the parameters are selected as $k = 0.4$, $W = 13$, $T_{area} = 1000$, $T_{size} = 1250$, and $r_{disk} = 9$.

4.3 Comparisons

We compare our results against those of the three marker identification algorithms. The first is the *intensity-based* algorithm. It defines the markers computing regional minima on the gray-level intensities I of the given image. Here, to alleviate the effects of noise, it uses the h -minima transform, which suppresses all minima in the intensity map I whose depth is less than a scalar h .

The second one is the *distance-based* algorithm which is similar to the intensity-based algorithm except that it uses the inverse of the distance transform instead of the intensity values. It obtains the distance transform map on the initial segmentation of an image such that the minimum distance from each foreground pixel to a background pixel is computed. Similarly, it uses the h -minima transform to reduce the effects of possible noise in the distance map. This algorithm necessitates obtaining an initial segmentation before finding the markers. For that, in our experiments, we use the cell regions that are identified at the end of the cell pixel quantization step as the initial segmentation; i.e., the union of the bright, dark, and dark-center pixels are used as the initial segmentation. Here we do not use the standard thresholding-based algorithms, which are typically used to obtain initial segmentations, since they yield worse results for our dataset.

The last is the *conditional-erosion* algorithm, which defines the markers on the initial segmentation map of the image by making use of iterative erosions [17]. It first iteratively erodes the connected components of the map with a coarse structuring element while the size of the components is greater than an area threshold. It repeats the same procedure on the resulting components, this time using a fine structuring element and a smaller area threshold. Likewise, we use the union of the bright, dark, and dark-center pixels identified by our algorithm as the initial segmentation.

These three algorithms have also their own model parameters. Besides, the method that we used to obtain their initial segmentation maps introduce additional parameters. In our experiments, we employ a similar method for the selection of these parameters: we first list different values for each parameter,

consider different combinations of these parameter values, and select the combination that yields the maximum F-score on the training cells.

4.4 Results

We present the quantitative results obtained by the proposed algorithm as well as the three comparison algorithms. As mentioned before, in marker-based evaluation, quantitative results are obtained using a distance threshold, which is used to find matches between the markers and the actual cells. Smaller values of this threshold increases false detections and misses since some of the identified markers are not close enough to the exact centroids of the gold standard cells. This decreases one-to-one matches, giving lower precision and recall values. Its larger values increases oversegmentations since more markers are matched to the same gold standard cell. This also decreases one-to-one matches. Figure 4.2 shows the number of one-to-one matches as a function of the distance threshold value for the training set. Considering these numbers, we select the distance threshold as 30, which gives the maximum one-to-one matches for all of the algorithms.

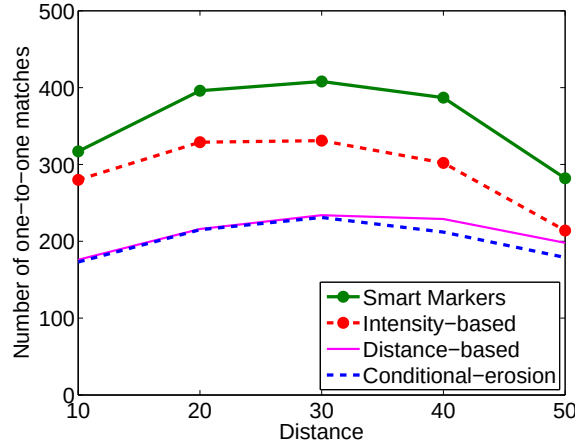


Figure 4.2: For the training set, the number of one-to-one matches as a function of the distance threshold value used in our distance-based evaluation algorithm.

We, first, present the quantitative results of marker-based evaluation for all of the algorithm. The results obtained on the training set are reported in Table 4.1

and Table 4.2 and on the test set in Table 4.3 and Table 4.4. The markers found by our proposed smart-markers algorithm as well as the comparison algorithms are visually shown on example images in Figure 4.3 and Figure 4.4.

Table 4.1: Comparison of the proposed smart markers algorithm against different marker identification algorithms. The results are obtained on the training set.

	One-to-one	Over.	Under.	False	Miss
Smart markers	408	4	10	29	53
Intensity-based	331	16	14	124	122
Distance-based	245	4	8	92	218
Conditional-erosion	231	23	8	97	226

Table 4.2: Comparison of the proposed smart markers algorithm against different marker identification algorithms, in terms of the *precision*, *recall* and *F-score* measures. The results are obtained on the training set.

	Precision	Recall	F-score
Smart markers	0.92	0.86	0.89
Intensity-based	0.70	0.70	0.70
Distance-based	0.71	0.52	0.60
Conditional-erosion	0.65	0.49	0.56

Table 4.3: Comparison of the proposed smart markers algorithm against different marker identification algorithms. The results are obtained on the test set.

	One-to-one	Over.	Under.	False	Misse
Smart markers	1284	36	52	122	129
Intensity-based	1102	50	50	138	309
Distance-based	834	17	34	240	604
Conditional-erosion	790	109	50	307	601

The results show that the definition of smart markers leads to higher precision and recall values. Compared to the other algorithms, it gives higher number of one-to-one matches with relatively less false detections and misses. In Table 4.1 - Table 4.4, we observe that the most successful comparison algorithm is the intensity-based algorithm. On the other hand, when we examine the visual results (Figure 4.3c and Figure 4.4c), we observe that this algorithm usually fails in finding Type I cells, which contains the bright pixels both in their centers and on their boundaries, and Type IV cells, which correspond to apoptosis. This is attributed to the following property of this algorithm: It identifies pixels with

Table 4.4: Comparison of the proposed smart markers algorithm against different marker identification algorithms in terms of the *precision*, *recall* and *F-score* measures. The results are obtained on the test set.

	Precision	Recall	F-score
Smart markers	0.88	0.87	0.87
Intensity-based	0.84	0.74	0.79
Distance-based	0.75	0.56	0.64
Conditional-erosion	0.64	0.53	0.58

lower intensities as markers. Thus, it cannot identify brighter pixels, which correspond to Type I cells, as markers. Besides, for images that contain noise and artifacts, it may find very large number of markers. Indeed, the reported results do not reflect this fact since we mask the markers with the initial segmentation found by our algorithm. If such a masking operation was not used, the number of false detections in the test set would increase from 138 to 427.

Moreover, the results show that the distance-based and conditional-erosion algorithms give less one-to-one matches due to the high number of misses. The visual results of the distance-based algorithm (Figure 4.3d and Figure 4.4d) and conditional-erosion algorithm (Figure 4.3e and Figure 4.4e) reveal that these algorithms are not successful in finding the clumped cells, regardless of their morphological classes. It is also worth noting that both of these algorithms require an initial segmentation² and the quality of this segmentation greatly affects the final segmentation results. In the experiments, we use the initial segmentation found by our proposed algorithm, which uses domain specific knowledge to define this segmentation. Without using this domain specific knowledge, it may be harder to find a good initial segmentation especially for Type III cells, which correspond to darker and non-circular cells, and Type IV cells, which correspond to apoptosis. This may further decrease the number of one-to-one matches.

We also present cell-based and area-based evaluations of the watershed algorithms, which grow the identified markers. In Table 4.5 and Table 4.7, we provide cell-based and area-based evaluation on the training set, respectively.

²The distance-based algorithm requires an initial segmentation to compute the distance map. The conditional-erosion algorithm requires it to conduct its iterative erosions.

Likewise, in Table 4.6 and Table 4.8, we report cell-based and area-based evaluation on the test set. Note that cell-based evaluation considers the correctly segmented cells after the flooding process of the watershed algorithms whereas marker-based evaluation considers those obtained before the flooding process. Thus, cell-based evaluation lead to one-to-one matches that are less than those computed in marker-based evaluation. Therefore, for all algrotihms, the *precision*, *recall*, and *F-score* measures computed by cell-based evaluation are smaller that those computed by marker-based evaluation.

Table 4.5: Cell-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the training set.

	Precision	Recall	F-score
Smart markers	0.83	0.77	0.80
Intensity-based	0.75	0.64	0.69
Distance-based	0.64	0.40	0.49
Conditional-erosion	0.58	0.38	0.46

Table 4.6: Cell-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the test set.

	Precision	Recall	F-score
Smart markers	0.84	0.83	0.84
Intensity-based	0.84	0.74	0.78
Distance-based	0.68	0.50	0.58
Conditional-erosion	0.61	0.47	0.53

Table 4.7: Area-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the training set.

	Precision	Recall	F-score
Smart markers	0.73	0.67	0.70
Intensity-based	0.71	0.54	0.62
Distance-based	0.50	0.38	0.43
Conditional-erosion	0.50	0.35	0.41

The quantitative results of cell-based and area-based evaluations reveal that the markers found by our proposed algorithm lead to finding better boundaries. The visual result results obtained after the flooding process are given on example

Table 4.8: Area-based evaluation of the marker-controlled watersheds that use the smart markers and those identified by the comparison algorithms. The results are obtained on the test set.

	Precision	Recall	F-score
Smart markers	0.80	0.72	0.76
Intensity-based	0.82	0.66	0.73
Distance-based	0.59	0.47	0.52
Conditional-erosion	0.58	0.44	0.50

images in Figure 4.5 and Figure 4.6. In these figures, we observe that the watershed algorithm that uses our smart markers give better results compared to the others. This is especially observed in the segmentation of clumped cells. Here, as consistent with the marker identification results, the intensity-based algorithm gives better segmentation compared to other comparison algorithms. This is attributed to the fact that the performance of a marker-controlled watershed algorithm highly depends on the success of the identified markers. On the other hand, the intensity-based algorithm uses low intensities to define its markers, it cannot identify the *Type I* cells, which mostly consist of bright pixels, as well as the *Type IV* cells, which show matte and irregular appearances. On the other hand, our proposed algorithm uses the knowledge of the existence of different cell types in an image and implements its marker identification and flooding process considering this knowledge. This is effective in improving the segmentation results.

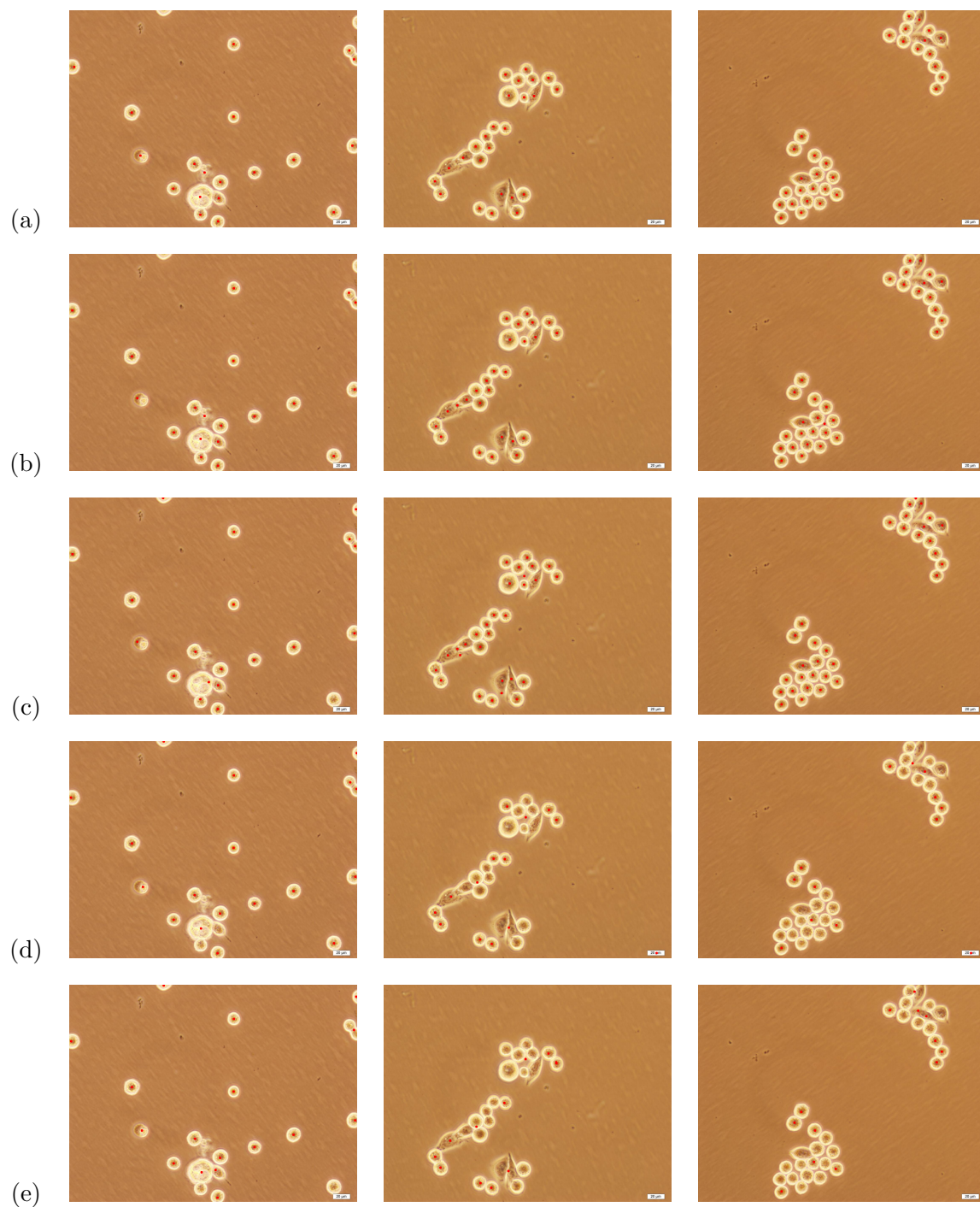


Figure 4.3: Visual results of the marker-identification algorithms, obtained on example images. (a) Gold standards and the markers found by (b) smart markers, (c) intensity-based, (d) distance-based, and (e) conditional-erosion algorithms.

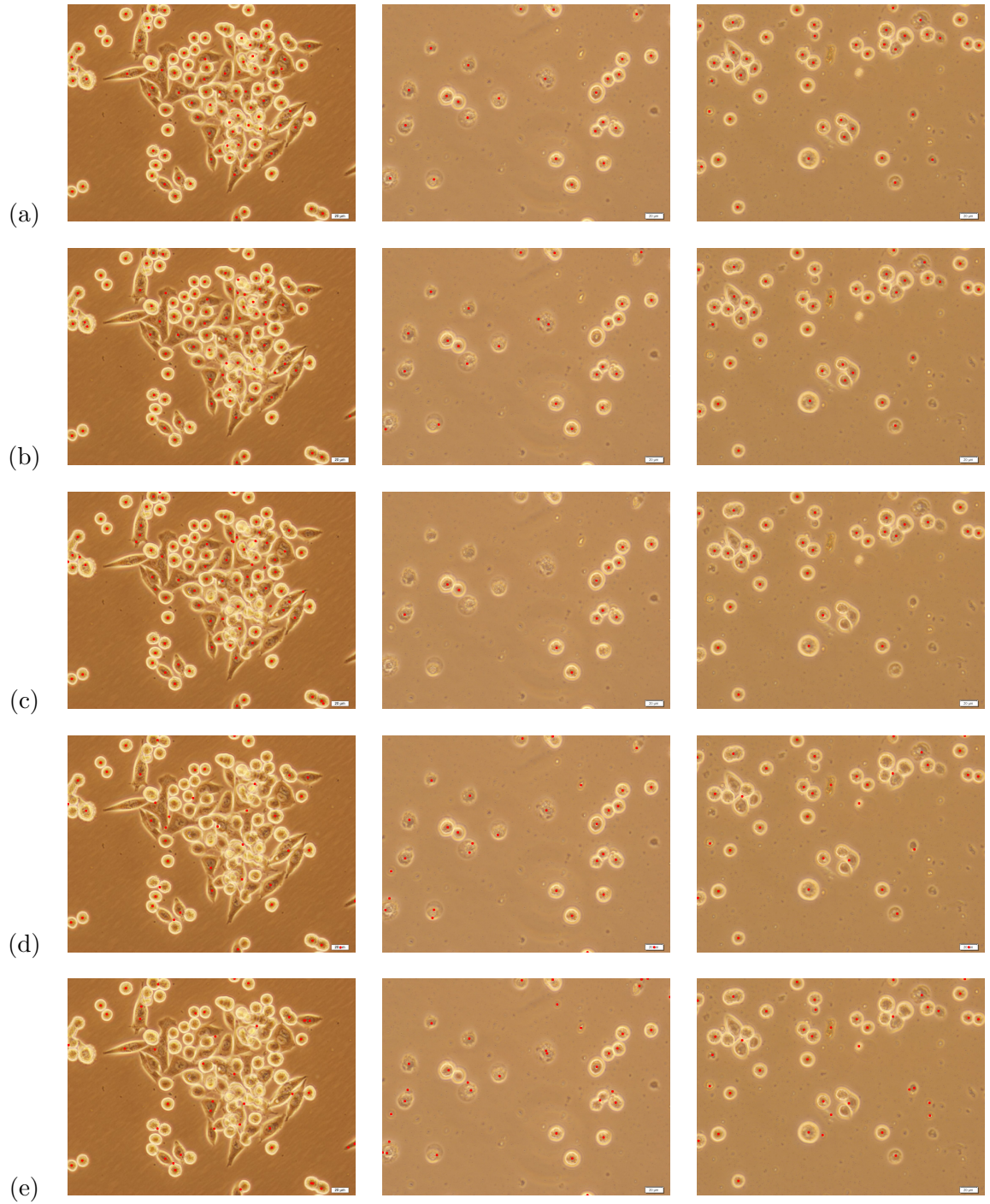


Figure 4.4: Visual results of the marker-identification algorithms, obtained on another set of example images. (a) Gold standards and the markers found by (b) smart markers, (c) intensity-based, (d) distance-based, and (e) conditional-erosion algorithms.

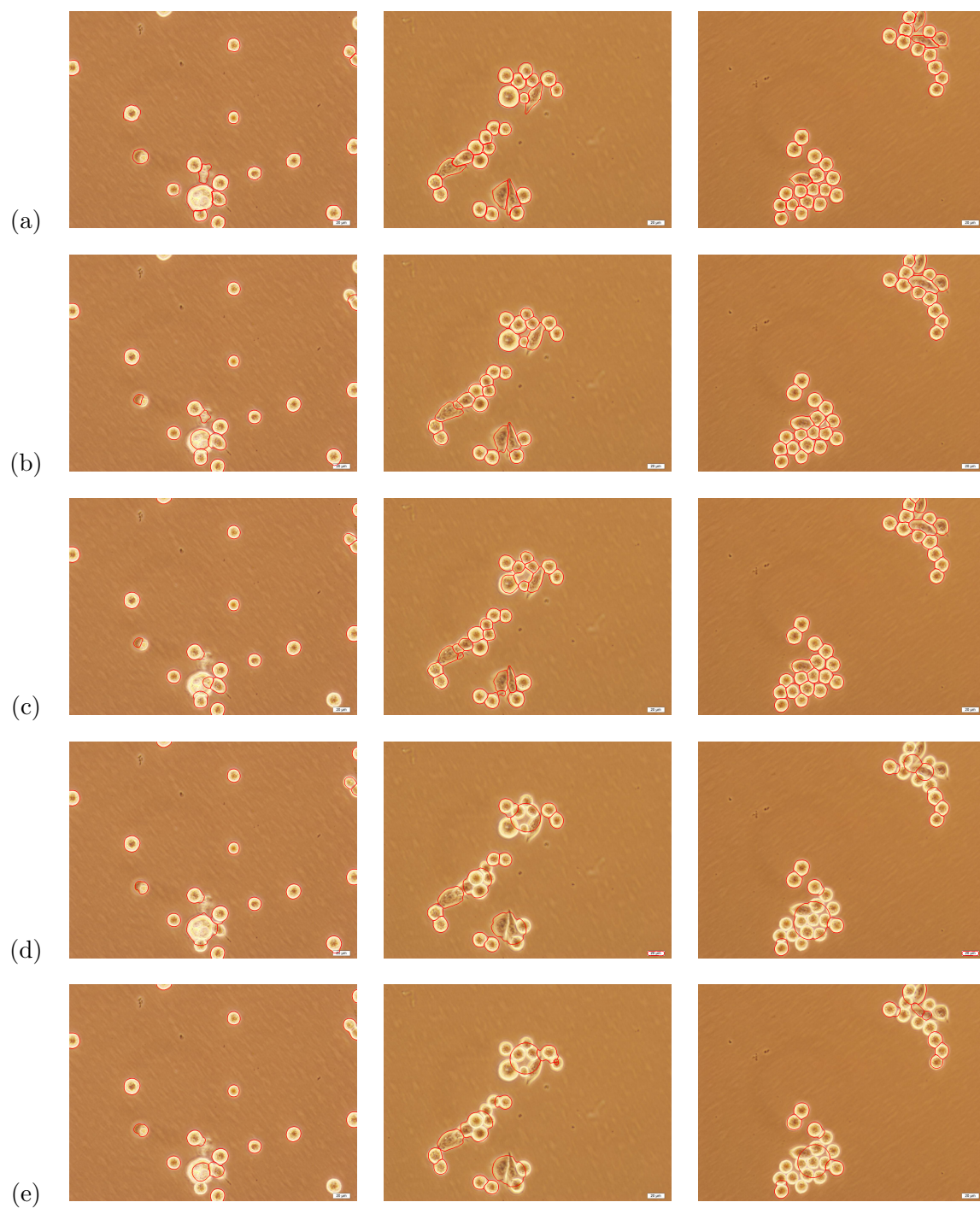


Figure 4.5: Visual results of the watershed algorithms, after growing the markers, obtained on example images. (a) Gold standards and the boundaries found by (b) smart markers, (c) intensity-based, (d) distance-based, and (e) conditional-erosion algorithms.

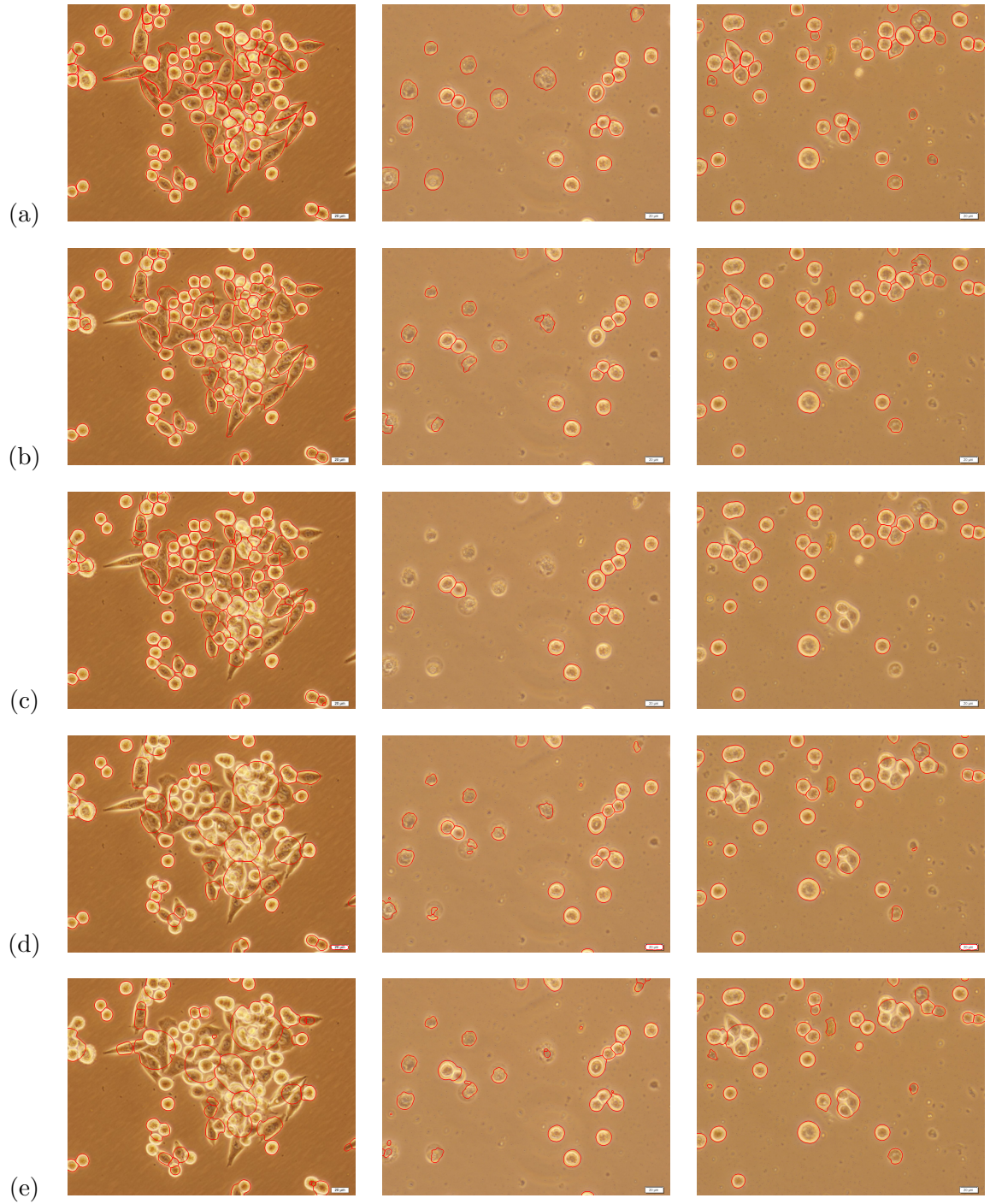


Figure 4.6: Visual results of the watershed algorithms, after growing the markers, obtained on another set of example images. (a) Gold standards and the boundaries found by (b) smart markers, (c) intensity-based, (d) distance-based, and (e) conditional-erosion algorithms.

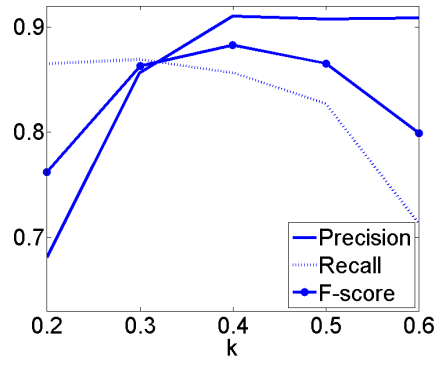
4.5 Parameter Analysis

The proposed algorithm has five model parameters: the Sobel threshold constant k , the filter size W , the area threshold T_{area} , the size threshold T_{size} , and the structuring element radius r_{disk} . In this section, we investigate the effects of each parameter to the segmentation performance. To this end, we fix four of these five parameters and observe the precision, recall, and F-score measures as a function of the other. In Figure 4.7 and Figure 4.8, we present the parameter analysis performed on the training and test set, respectively.

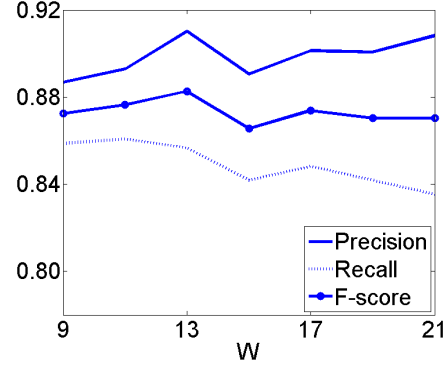
There are three external parameters in the cell pixel quantization step. The first one is the Sobel threshold constant k that is used to define dark pixels. When its smaller values are used, some background pixels are also defined as dark so that false background regions are identified as cells. This increases the number of computed cells without increasing one-to-one matches, which in turn lowers precision. On the other hand, when larger values of this constant are used, less dark pixel components can be found. This leads to less computed cells as well as less one-to-one matches, which lowers recall. Note that larger values do not lower precision since the number of computed cells and one-to-one matches decrease concurrently. In our experiments, this parameter is selected as 0.4. Figure 4.7a and Figure 4.8a shows that this selected value provides a good balance between precision and recall.

The second parameter of this step is the size W of the majority filter that is used for alleviating the effects of noise in pixel quantization. The filter size W should be selected large enough to get the benefits of majority filtering. On the other hand, selecting too large filter sizes causes to assign incorrect labels to pixels. As seen in Figure 4.7b and Figure 4.8b, this changes the balance between precision and recall. The area threshold T_{area} is the last parameter of this step. It is used to eliminate smaller dark components. Smaller threshold values identify more false regions as cells whereas larger values give less computed cells and one-to-one matches. These decrease precision and recall, respectively, as in the case of the parameter k . In the experiments, T_{area} is selected to be 1000, which gives high precision and recall values at the same time (Figure 4.7c and Figure 4.8c).

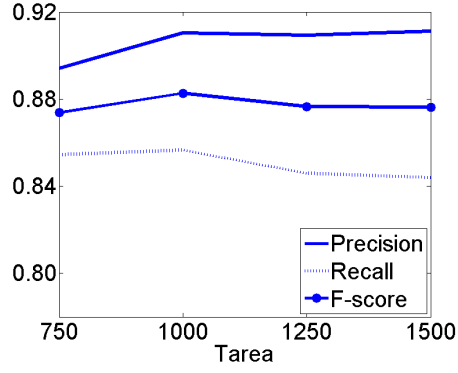
There are two parameters used in the smart marker extraction step. These are the size threshold T_{size} and the radius r_{disk} of the structuring element. Smaller values of T_{size} cause to define false markers, increasing the number of computed cells without changing one-to-one matches. On the other hand, its larger values cause to eliminate some true markers, decreasing the number of computed cells as well as one-to-one matches. These two conditions decrease precision and recall values, respectively, as observed in Figure 4.7d and Figure 4.8d. The radius r_{disk} slightly changes the results except the case when largest values are used (Figure 4.7e and Figure 4.8e). The largest values prevent the iterative erosion algorithm to identify especially smaller true markers; this lowers recall as well. In the experiments, $T_{size} = 1250$ and $r_{disk} = 9$, which give a good balance between precision and recall.



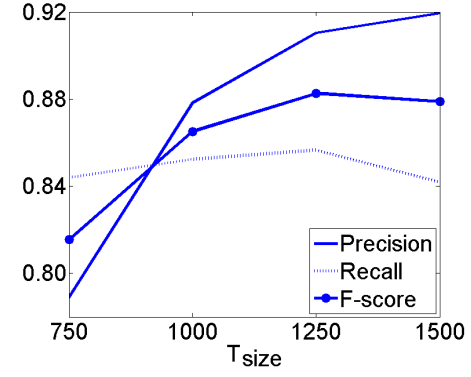
(a)



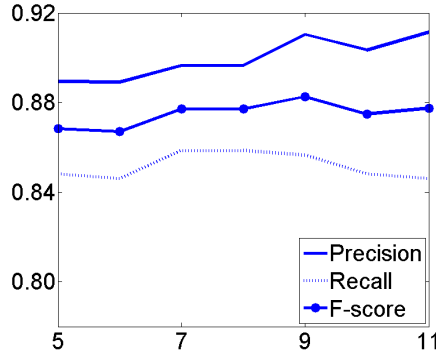
(b)



(c)

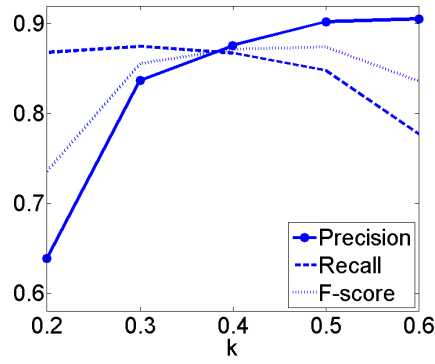


(d)

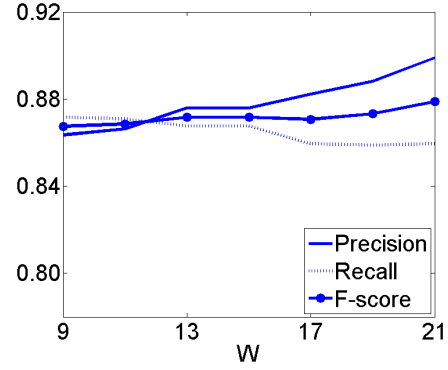


(e)

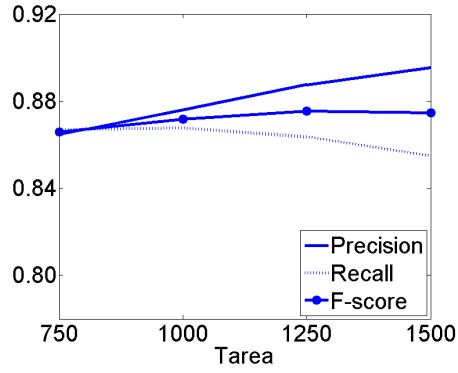
Figure 4.7: For the training set, the precision, recall, and F-score measures as a function of (a) the Sobel threshold constant k , (b) the size W of the majority filter, (c) the area threshold T_{area} , (d) the size threshold T_{size} , and (e) the radius r_{disk} of the structuring element.



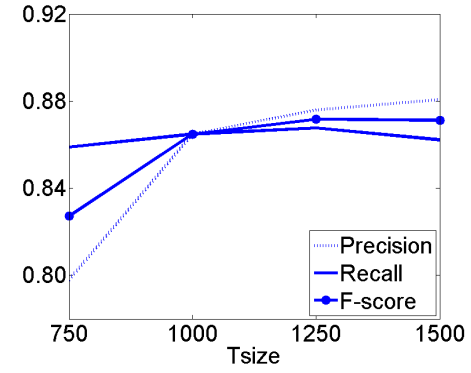
(a)



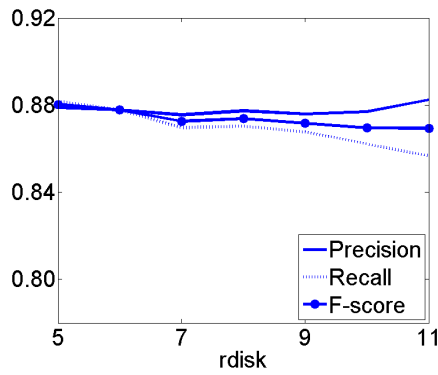
(b)



(c)



(d)



(e)

Figure 4.8: For the test set, the precision, recall, and F-score measures as a function of (a) the Sobel threshold constant k , (b) the size W of the majority filter, (c) the area threshold T_{area} , (d) the size threshold T_{size} , and (e) the radius r_{disk} of the structuring element.

Chapter 5

Conclusion

In this work, we introduced the idea of defining smart markers for a marker-controlled watershed algorithm by making use of domain knowledge specific to live cells. This definition relies on defining different pixel groups based on the morphological characteristics of the live cells and identifying the smart markers on these pixel groups. Working with 1954 KATO-3 gastric cancer cells, our experiments indicated the effectiveness of this smart marker definition in obtaining more successful results.

As seen in the visual results, the proposed algorithm can successfully find different types of cells. This is attributed to the fact that the algorithm uses the domain specific knowledge so that it knows there exist different types of cells in a cell line and the characteristics of these cells. Therefore, it can use this knowledge in defining its markers. On the other hand, the other algorithms do not use the knowledge of the existence of different cell types in a cell line (or a tissue). The ability of using such knowledge is indeed closely related with working on live cells. The live cells are not fully attached to the plate, and thus, the cells belonging to different morphological classes can show different appearances. On the other hand, when the cells are fixed, they become fully attached to the plate and their appearances become the same. The only exception is the appearance of dead (e.g., apoptotic) cells; they usually seem different than the others. Thus, to analyze the morphological classes of the fixed cells, special stainings are typically

required. Most of the algorithms in literature, including those that we used in our comparisons, were implemented considering the fixed cells (mostly for fluorescence stained cells). This could be the reason of these algorithms not considering such kind of knowledge in their segmentations. Our proposed work is a good example of showing how the domain knowledge can effectively be used in a cell segmentation algorithm.

We developed our algorithm considering the morphological characteristics of the KATO-3 human gastric cancer cell line. We also use the images of this cell line to test our algorithm. Nevertheless, it is also possible to apply this algorithm to other cell lines. To explore this possibility, we also test our algorithm on four different cell lines, namely the Huh7 human liver cancer, MCF7 human breast cancer, MFE-296 human endometrial carcinoma, and SK-BR-3 human breast cancer cell lines. The preliminary visual results obtained on example images of these cell lines are given in Figure 5.1. This figure shows that the results hold promise for the proposed algorithm to be also used for different cell lines. In order to obtain better results, one can consider the characteristics of these cell lines for the definition of additional pixel groups as well as for the identification of additional smart marker types on these pixel groups. This could be considered as one of the future research directions of our work.

The quantitative results reveal that the use of the proposed smart markers gives more successful results than the other algorithms in both area-based and cell-based evaluations, both of which are obtained on the grown markers. When area-based and cell-based evaluation results are considered together, one can observe that the watershed algorithm that uses the smart markers identifies cells better than finding their exact boundaries. To improve the segmented boundaries, one can combine different criteria, such as intensity and gradient values, with the pixel groups in the growing process. This would be another future research direction of this work.

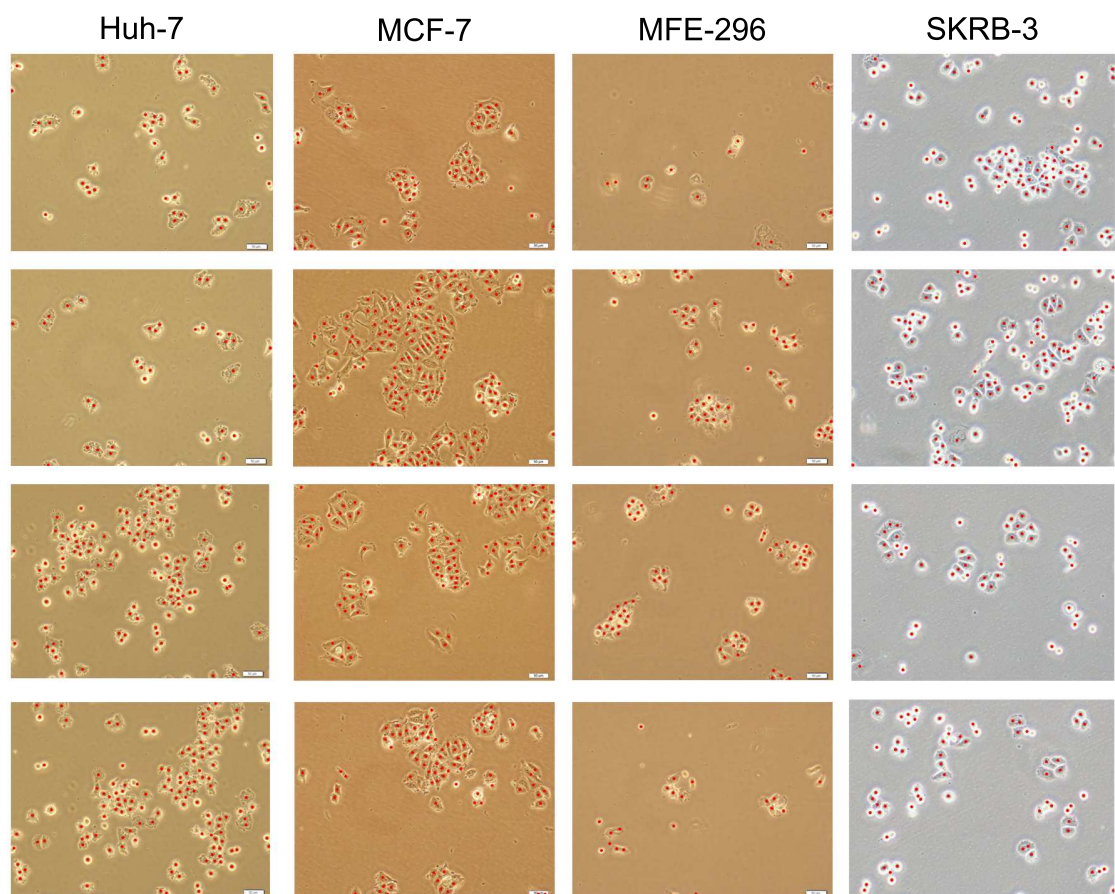


Figure 5.1: Visual results of the proposed algorithm obtained on the images of different cell lines.

Bibliography

- [1] X. Chen, X. Zhou, and S. Wong, “Automated segmentation, classification, and tracking of cancer cell nuclei in time-lapse microscopy,” *Biomedical Engineering, IEEE Transactions on*, vol. 53, 2006.
- [2] H. Zhou and K. Mao, “Adaptive successive erosion-based cell image segmentation for p53 immunohistochemistry in bladder inverted papilloma,” in *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pp. 6484–6487, 2005.
- [3] N. Harder, F. Mora-Bermúdez, 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,” in *Proceedings of the 9th international conference on Medical Image Computing and Computer-Assisted Intervention - Volume Part I, MICCAI’06*, pp. 840–848, 2006.
- [4] G. Kayser, D. Radziszowski, P. Bzdyl, R. Sommer, and K. Kayser, “Theory and implementation of an electronic, automated measurement system for images obtained from immunohistochemically stained slides,” *Anal. Quant. Cytol. Histol.*, vol. 28, no. 1, pp. 27–38, 2006.
- [5] K. Kayser, H. Schultz, T. Goldmann, J. Gortler, G. Kayser, and E. Vollmer, “Theory of sampling and its application in tissue based diagnosis,” *Diagn Pathol*, vol. 4, 2009.
- [6] M. Park, J. S. Jin, M. Xu, W. S. F. Wong, S. Luo, and Y. Cui, “Microscopic image segmentation based on color pixels classification,” in *Proceedings of*

the First International Conference on Internet Multimedia Computing and Service, ICIMCS '09, pp. 53–59, 2009.

- [7] 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 Trans Med Imaging*, vol. 21, pp. 1212–1221, 2002.
- [8] B. Zhang, C. Zimmer, and J. C. Olivo-Marin, “Tracking fluorescent cells with coupled geometric active contours,” in *Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on*, pp. 476–479, 2004.
- [9] G. Xiong, X. Zhou, and L. Ji, “Automated segmentation of drosophila rna1 fluorescence cellular images using deformable models,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, pp. 2415–2424, 2006.
- [10] N. Malpica, C. O. de Solorzano, J. J. Vaquero, A. Santos, I. Vallcorba, J. M. Garcia-Sagredo, and F. del Pozo, “Applying watershed algorithms to the segmentation of clustered nuclei,” *Cytometry*, vol. 28, pp. 289–297, 1997.
- [11] 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 A*, vol. 73, pp. 958–964, 2008.
- [12] 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, pp. 838–853, 2011.
- [13] J. Lindblad, C. Wahlby, E. Bengtsson, and A. Zaltsman, “Image analysis for automatic segmentation of cytoplasm and classification of Rac1 activation,” *Cytometry A*, vol. 57, pp. 22–33, 2004.
- [14] M. Wang, X. Zhou, F. Li, J. Huckins, R. W. King, and S. T. Wong, “Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy,” *Bioinformatics*, vol. 24, pp. 94–101, 2008.

- [15] J. Cheng and J. C. Rajapakse, "Segmentation of clustered nuclei with shape markers and marking function," *IEEE Trans Biomed Eng*, vol. 56, pp. 741–748, 2009.
- [16] C. Jung and C. Kim, "Segmenting clustered nuclei using H-minima transform-based marker extraction and contour parameterization," *IEEE Trans Biomed Eng*, vol. 57, pp. 2600–2604, 2010.
- [17] X. Yang, H. Li, and X. Zhou, "Nuclei segmentation using marker-controlled watershed, tracking using mean-shift, and kalman filter in time-lapse microscopy," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, no. 11, pp. 2405–2414, 2006.
- [18] C. Jung, C. Kim, S. W. Chae, and S. Oh, "Unsupervised segmentation of overlapped nuclei using Bayesian classification," *IEEE Trans Biomed Eng*, vol. 57, no. 12, pp. 2825–2832, 2010.
- [19] P. U. Adiga and B. Chaudhuri, "An efficient method based on watershed and rule-based merging for segmentation of 3-d histo-pathological images," *Pattern Recognition*, vol. 34, pp. 1449–1458, 2001.
- [20] C. Wahlby, 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," *J Microsc*, vol. 215, pp. 67–76, 2004.
- [21] 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 A*, vol. 63, pp. 20–33, 2005.
- [22] X. Zhou, F. Li, J. Yan, and S. T. Wong, "A novel cell segmentation method and cell phase identification using Markov model," *IEEE Trans Inf Technol Biomed*, vol. 13, pp. 152–157, 2009.
- [23] E. Bak, K. Najarian, and J. P. Brockway, "Efficient segmentation framework of cell images in noise environments," *Conf Proc IEEE Eng Med Biol Soc*, vol. 3, pp. 1802–1805, 2004.

- [24] F. Liu, D. Xu, M. S. Ferguson, B. Chu, T. Saam, N. Takaya, T. S. Hatsukami, C. Yuan, and W. S. Kerwin, "Automated in vivo segmentation of carotid plaque MRI with Morphology-Enhanced probability maps," *Magn Reson Med*, vol. 55, pp. 659–668, 2006.
- [25] K. Tingelhoff, A. I. Moral, M. E. Kunkel, M. Rilk, I. Wagner, K. G. Eichhorn, F. M. Wahl, and F. Bootz, "Comparison between manual and semi-automatic segmentation of nasal cavity and paranasal sinuses from CT images," *Conf Proc IEEE Eng Med Biol Soc*, pp. 5505–5508, 2007.
- [26] N. Kharma, H. Moghnieh, J. Yao, Y. P. Guo, A. Abu-Baker, J. Laganier, G. Rouleau, and M. Cheriet, "Automatic segmentation of cells from microscopic imagery using ellipse detection," *Image Processing, IET*, vol. 1, pp. 39–47, 2007.
- [27] M. Sezgin and B. Sankur, "Survey over image thresholding techniques and quantitative performance evaluation," *Journal of Electronic Imaging*, vol. 13, pp. 146–168, 2004.
- [28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 9, pp. 62–66, 1979.
- [29] M. E. Plissiti, A. Charchanti, O. Krikoni, and D. I. Fotiadis, "Automated segmentation of cell nuclei in pap smear images," in *Proceedings of IEEE International Special Topic Conference on Information Technology in Biomedicine*, pp. 6484–6487, 2006.
- [30] F. Yang and T. Jiang, "Cell image segmentation with kernel-based dynamic clustering and an ellipsoidal cell shape model," *J Biomed Inform*, vol. 34, pp. 67–73, 2001.
- [31] T. Chen, Y. Zhang, C. Wang, Z. Qu, and S. T. C. Wong, "Neural stem cell segmentation using local complex phase information," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pp. 3637–3640, 2010.
- [32] J. M. White and G. D. Rohrer, "Image thresholding for optical character recognition and other applications requiring character image extraction," *IBM J. Res. Dev*, vol. 27, pp. 400–411, 1983.

- [33] S. Kumar, S. H. Ong, S. Ranganath, F. T. Chew, and T. C. Ong, "Segmentation of microscope cell images via adaptive eigenfilters," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, vol. 1, pp. 135–138, 2004.
- [34] P. W. Palumbo, P. Swaminathan, and S. N. Srihari, "Document image binarization: Evaluation of algorithms," in *Proc. SPIE, Applications of Digital Image Processing IX*, vol. 697, pp. 278–285, 1986.
- [35] S. Kothari, Q. Chaudry, and M. D. Wang, "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques," in *Proceedings of the Sixth IEEE international conference on Symposium on Biomedical Imaging: From Nano to Macro*, pp. 795–798, 2009.
- [36] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, 1988.
- [37] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Trans Image Process*, vol. 10, pp. 266–277, 2001.
- [38] M. Tscherepanow, F. Zillner, M. Hillebrand, and F. Kummert, "Automatic segmentation of unstained living cells in bright-field microscope images," in *Advances in Mass Data Analysis of Images and Signals in Medicine, Biotechnology, Chemistry and Food Industry*, vol. 5108, pp. 158–172, 2008.
- [39] S. Beucher and C. Lantuejoul, "Use of watersheds in contour detection," in *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France.*, 1979.
- [40] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. 2001.
- [41] X. Bai, C. Sun, and F. Zhou, "Splitting touching cells based on concave points and ellipse fitting," *Pattern Recognition*, vol. 42, pp. 2434–2446, 2009.
- [42] 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.

- [43] A. B. Tosun, M. Kandemir, C. Sokmensuer, and C. Gunduz-Demir, “Object-oriented texture analysis for the unsupervised segmentation of biopsy images for cancer detection,” *Pattern Recognition*, vol. 42, pp. 1104–1112, 2009.