Segmentation of Bright-field Cell Images

by

Laura Dianne Bradbury

A research paper
presented to the University of Waterloo
in fulfillment of the
project requirement for the degree of
Master of Mathematics
in
Computational Mathematics

Supervisor: Justin W.L. Wan

Waterloo, Ontario, Canada, 2009

I hereby declare that I am the sole author of this research paper. This is a true copy of the research paper, including any required final revisions, as accepted by my examiners.

I understand that my research paper may be made electronically available to the public.

Abstract

Image segmentation is a complex problem with many practical applications. In particular segmenting images of cells and tracking each cell through a series of images has the potential to increase the throughput of cell experiments. Bright-field images present many challenges to segment due to their poor contrast with the image background, broken boundaries, partial halos and overlapping cells. In this paper I present three solutions for the bright-field cell image segmentation problem. The first solution is based on a multiphase level set segmentation technique and the watershed method, the second is based on the minimization of a statistical energy function by using a level set function, while the third is based on K-Means segmentation, spectral partitioning and separation of variances. The first technique and the third technique produce good segmentation results of individual cell images, as well as fit into the proposed multiple cell segmentation framework.

Acknowledgements

I would to thank Justin W.L. Wan for all of his guidance with this project. I would also like to thank Thomas Hudson of the Ontario Institute for Cancer Research and Robert Sladek and Haig Djambazian of the Department of Medicine and Human Genetics, McGill University and Genome Quebec Innovation. Finally I would like to thank Sarav Sundarajan for generating the bright-field cell images that are used in this paper.

Dedication

I dedicate this paper to my family.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Image segmentation is a key part of many applications. It allows traffic cameras to read license plates, it allows cameras in casinos to monitor play, it can be used to detect masses in breast tissue or the prostate and it can also be used by biologists to locate the boundary of cells in an image. Cell biologists run hundreds of experiments on thousands of cells, experiments to learn more about how our bodies work and why diseases like cancer and aids occur. These experiments often involve an image being acquired at regular intervals in time. A single experiment can result in hundreds of images, each containing hundreds, if not thousands of cells. Currently many of these images are processed by hand. That is, given a sequence of images, a technician must track each cell though the images, noting how fast it moves, how often it divides, its volume and any other pertinent details. The focus of this project is to develop a technique that can segment bright-field images of cells to be part of an automatic tracking algorithm for a series of images.

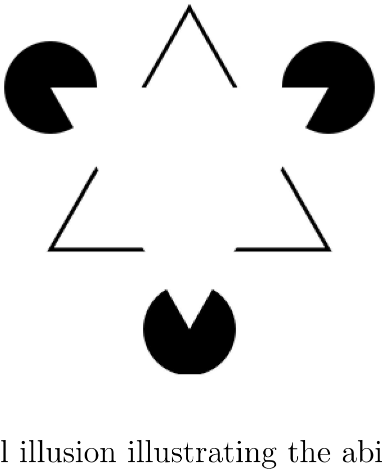

Figure 1.1: An optical illusion illustrating the ability of the human eye

Image segmentation is the process of locating the boundaries of objects in an image. This is easy for the human eye. The human eye and brain can process huge amounts of information almost instantaneously. Humans have the ability to infer information from images, to see edges that are not really there and to complete a broken boundary in a natural way. For example, in Figure 1.1, we easily see the white triangle in the image, despite the fact that most of its boundary is missing. Computer vision algorithms are no where close to matching the natural ability of the human eye, and therefore tasks that are simple for a human to complete are extremely difficult to automate. The solutions presented in this paper use techniques primarily based on the intensity values of each pixel in the image to segment bright-field images of cells.

1.1 Eukaryotic Cells

There are two types of cells, prokaryotic and eukaryotic, which differ due to their size and internal structures. Prokaryotic Cells are simpler cells such as bacteria cells, while eukaryotic cells include plant and animal cells [9]. This project concerns the imaging of eukaryotic cells.

The (eukaryotic) cell is a transparent object that contains different organelles and structural elements suspended in the cell's cytoplasm. The cytoplasm is essentially the body of the cell, it is enclosed in a plasma membrane (a semipermeable lipid bilayer). In eukaryotic cells, the cytoplasm contains organelles such as mitochondria and ribosomes, which are all separated from the rest of the cytoplasm by biological membranes. The nucleus is also considered to be an organelle. It is separated from the rest of the cell by a semi permeable membrane and contains the genetic material, deoxyribonucleic acid (DNA), for the cell. A typical human cell, while only $10\mu m$ contains nearly $2.5m$ of DNA. The cell also changes its appearance rapidly at times. Most types of cells progress through the cell cycle, which includes reproduction through mitosis. Mitosis is process of cell division and includes five stages: Prophase, Pro-metaphase, Metaphase, Anaphase and Telophase. During this time the form of the DNA is changing, and duplicating, and the cell changes shape to assist, becoming small and round. Finally the cell pulls apart during cytokinesis, resulting in a figure eight shape until the two daughter cells separate. The new daughter cells then enter interphase where the cell is often spread out and irregular in shape [9].

1.2 Bright-field versus Fluorescent Microscopy

There are many types of microscopy used by cell biologists [9]. Bright-field microscopy is the simplest of all optical microscopy techniques. The sample is illuminated from below using white light. Fluorescent microcopy is more complicated. The cells are tagged with a compound that will fluoresce under certain light conditions. This means that a gene to produce a fluorescent protein may be introduced into the genome of the cell, tagged DNA markers may be used or the proteins that the cell needs to function may include amino acids that are tagged to be fluorescent. The fluorescent cells will emit a certain wavelength of light. Images are taken to capture this specific wavelength. Both imaging types are important to cell biologists.

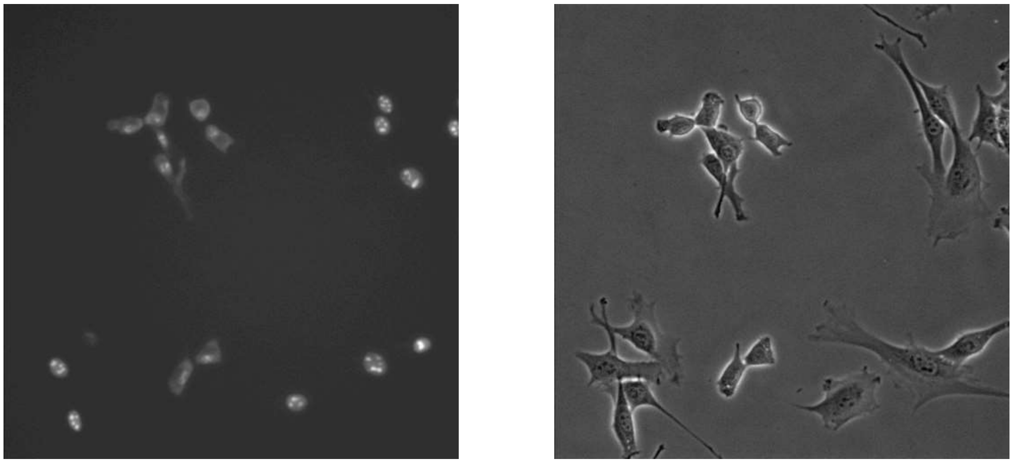

Figure 1.2: The same cells with different imaging types (Left: Fluorescent, Right: Bright-field)

There are advantages and disadvantages to each type of microscopy. Images from fluorescent microscopy are easily segmented [18]. This is because the region of interest is a bright, convex shape that is well separated from the background and the objects of interest do not overlap, as shown in Figure 1.2 (left). However, fluorescent images often capture only the nucleus of the cell. If the experiment requires data such as the area of the cell than this type of imaging will not work alone. Furthermore, it is possible that the agent used to cause the cell to fluoresce will change the behaviour of the cell, i.e. how fast the cell grows and divides, or how fast the cell moves. Another known issue with fluorescent imaging is that the cell may disappear for a few frames and then reappear. This can cause problems when a cell reappears since the biologists do not know if the new cell is the same cell from before or the result of cell division. Images from bright-field microscopy provide a much better image of the cell, as it includes both the cytoplasm and the nucleus which provide more information for the biologists, which can be seen in Figure 1.2 (right). However the image quality creates problems for segmentation, the overall intensity of the background is approximately the same as the intensity of the cell. Also, since the image is formed from light transmission, the gray level of each pixel incorporates information form all depths of the cell in that location [10]. The cytoplasm absorbs some of the light, and the organelles absorb light to various higher degrees than the cytoplasm. For this reason the cell body is not uniform in intensity in the grey-level images. The brighter spots occur because some of the cellular structures behave as converging lenses or there is light reflecting on the cell surface. Furthermore the cell often as a bright white area surrounding it know as a halo. Unfortunately the halo is not easily used to produce a segmentation as it is often broken or can be missing altogether. The cell itself often has a poorly defined boundaries, which makes edge stopping techniques for segmentation difficult to use for this application. The dark outline is present due to light reflection or distortion on as light passes through the part of the cell membrane that is not parallel to the plane. The parts of the cell that have poor boundaries with the background occur because the slope of the cell membrane is gentle, thus round cells have good contrast with the background, while flattened cells have a poorer contrast. Another known issue is the cytoplasm of cells overlaps so cells are often occluded. Finally, the cell can change appearance rapidly as it moves through the cell cycle so it is not possible to search for the image of the cell from the previous frame in the new frame.

The segmentation problem for fluorescent images has been solved, many different approaches are used with a strong degree of success. These methods do not work well for the corresponding bright-field cell images because of the different nature of the images. These algorithms and their results will be discussed further in Chapter 3.

1.3 Problem to be Solved

The goal of this project is to be able to segment images of bright-field cells that are close together (but not overlapping) and cells that divide. The final version of the algorithm should be adjustable by means of parameters to other image sets, but for a single set of images form a single experiment, the parameters should not require adjustment. The first image in a sequence may require some manual work to start the segmentation, but from that point on the algorithm should run automatically without human input. This problem is designed to be offline, that is not completed in real time, so time constrains will not be considered at this time.

1.4 Remainder of Paper

The remainder of the paper is organized as follows: Chapter 2 contains the background information on a variety of common segmentation and clustering algorithms; Chapter 3 contains a literature review of other cell segmentation algorithms; Chapter 4 contains a description of the algorithms which have results in this paper; Chapter 5 has an algorithm for a multiple cell segmentation; Chapter 6 has the results for all algorithms presented and finally Chapter 7 is the conclusion of this paper.

Chapter 2

Background

There are many approaches to solve segmentation problems: level set methods rely on PDE's, spectral clustering is based on graph theory and eigenvalues, while the watershed method treats the image as a topography. This chapter discusses some active contour methods for segmentation, the watershed method, spectral clustering and the K-Means algorithm.

2.1 Level Set Segmentation

Level set methods were introduced by Osher [12] and Sethian [13]. The goal of a level set segmentation method is to find a curve that describes the boundary of the object of interest. This is accomplished in 2D by a propagating a 3D function. The zero level set of the the 3D function describes the 2D boundary of the object of interest. Level set methods are used for applications such as image segmentation, optimization, computational geometry and computational fluid dynamics.

In image segmentation, level set methods are part of a class of methods known as active contour methods. Another active contour method is called snakes. In the snakes method the curve is represented as a parametrized set of points, as the points move, so does the curve. Level set methods have an advantage over other active contour methods such as snakes because a level set method can automatically split or joint regions. Also complex topologies such as three way junctions and rings are easily managed with a level set method, however with a method like snakes it is difficult to know when to split or join regions, or if it is necessary to introduce a new boundary.

The basic idea for any active contour method (snakes or level set for instance) is to evolve a curve over time so that the curve moves towards its interior normal and when the stopping conditions are met the curve forms an outline around the object of interest. If Ω is an open bounded set in $\mathbb{R}^2$ with a boundary $\partial\Omega$ then the level set function ϕ evolves such that the set $\{(x, y)|\phi(x, y) = 0\}$ defines $\partial\Omega$. The

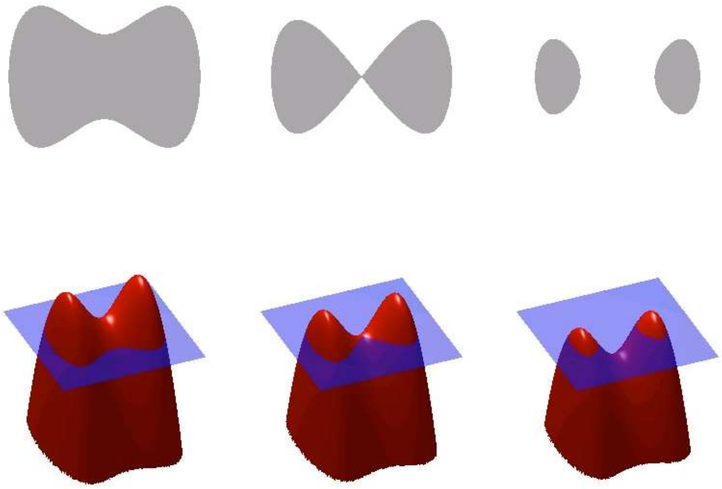

Figure 2.1: Example of a 3D Level Set Function giving a 2D curve[1].

general level set equation is given by:

$$\frac{\partial \phi}{\partial t} = F|\nabla \phi|,\ \phi(0,x,y) = \phi_0(x,y), \tag{2.1}$$

where F is the speed term, which dictates how fast the curve moves at each point in the image, while ϕ_0 is the zero level set of the function ϕ and represents the boundary of the current segmentation. Figure 2.1 shows how the zero level set of the function ϕ forms the curve in 2D.

The level set function ϕ has the following properties:

$$\begin{aligned} \phi(\mathbf{x},t) > 0 \quad & \text{for } \mathbf{x} \in \Omega \\ \phi(\mathbf{x},t) < 0 \quad & \text{for } \mathbf{x} \in \mathbb{R}^n \backslash \Omega \\ \phi(\mathbf{x},t) = 0 \quad & \text{for } \mathbf{x} \in \partial\Omega = \Gamma(t) \end{aligned}$$

The following are terms that are useful for describing level set methods:

1. The unit normal N to $\partial\Omega$ is given by $\frac{\nabla\phi}{|\nabla\phi|}$
2. The mean curvature κ of $\partial\Omega$ is defined by $\kappa = -\nabla\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$
3. The one-dimensional Heaviside function $\begin{cases} H(x) \equiv 1 & \text{if } x > 0 \\ H(x) \equiv 0 & \text{if } x < 0 \end{cases}$
4. The one-dimensional dirac delta function $\delta(x) = \begin{cases} +\infty & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases}$

[1] Source: Wikimedia Commons, Oleg Alexandrov, 2004

The solution of (2.1) usually develops kinks or discontinuities. Because of these discontinuities special numerical methods to calculate the solution must be used. For example the schemes generally need to be monotonic and use up-wind differencing, otherwise numerical instability appears.

The level set equation (2.1) can be used for image segmentation by defining F as in the anisotropic diffusion approach:

$$\frac{\partial \phi}{\partial t} = g(|\nabla u_0|)|\nabla \phi|(\mathrm{div}(\frac{\nabla \phi}{|\nabla \phi|}) + \nu), \tag{2.2}$$

where the function g is a function that is large when $|\nabla u_0|$ is small and large when $|\nabla u_0|$ is small. By using this as the speed term the curve will stop when it reaches a sharp change in pixel intensity (an edge). Finally ν is a constant that is part of the the curvature term.

Figure 2.2 shows a level set function evolving from its initial state to its final position. This image uses (2.2) to propagate the curve. The speed term used is $g(|u_0|) = \frac{1}{1+|\nabla u_0|_2^2}$. The PDE is solved using a finite difference method with an up-winding scheme.

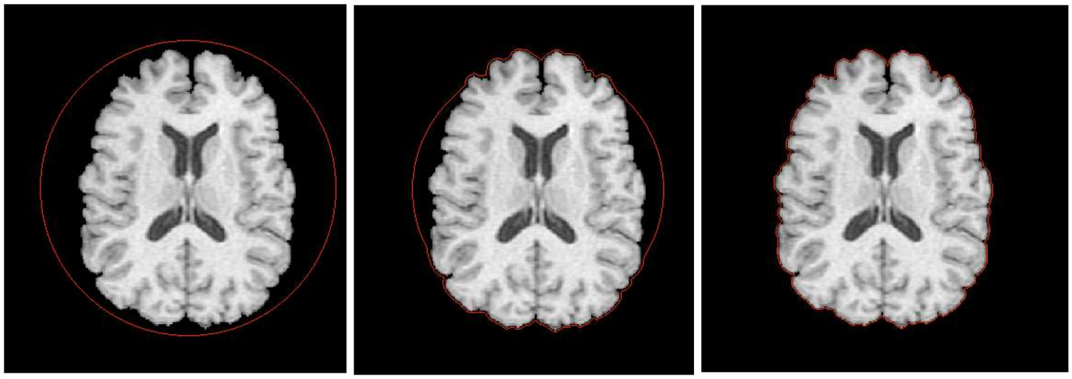

Figure 2.2: Example of a level set segmentation based on the anisotropic diffusion model

2.1.1 Active Contours Without Edges

In [4], Chan and Vese propose a new model for active contours to detect objects in a given image. The method is based on of the Mumford-Shah functional for segmentation and level sets. The method uses minimization of energy to segment the image. In the simple case, assume that an image u_0 is formed from two regions of approximately piecewise constant intensities of distinct values u_0^{in} and u_0^{out}. Furthermore assume that the intensity of the object to be detected is u_0^{in}. If C_0 is the boundary of the object to be detected then $u_0 \approx u_0^{in}$ inside the object (or inside

C_0) and $u_0 \approx u_0^{out}$ outside the object (or outside C_0). The fitting terms considered are:

$$\begin{aligned} F_1(C) + F_2(C) &= \int_{inside(C)} (u_0(x,y) - c_1)^2 dxdy \\ &\quad + \int_{outside(C)} (u_0(x,y) - c_2)^2 dxdy. \end{aligned} \tag{2.3}$$

In (2.3), C is a variable curve on the image, the constants c_1 and c_2 are the mean intensities of u_0 inside and outside C respectively. In this case it is clear finding the boundary of the object is the same as finding the curve C that minimizes the fitting term:

$$\inf_C \{F_1(C) + F_2(C)\} \approx 0 \approx F_1(C_0) + F_2(C_0). \tag{2.4}$$

Figure 2.3 shows how the curve minimizes (2.4) .

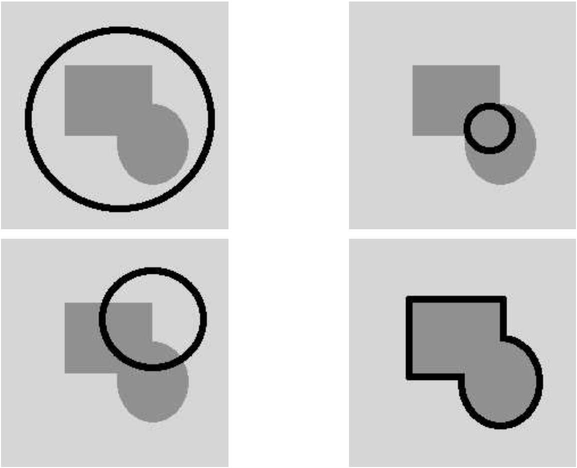

Figure 2.3: Four examples showing how the curve minimizes (2.4), [4]. Top left: $F_1(C) > 0$, $F_2(C) \approx 0$, Fitting > 0. Top right: $F_1(C) \approx 0$, $F_2(C) > 0$, Fitting > 0. Bottom left: $F_1(C) > 0$, $F_2(C) > 0$, Fitting > 0. Bottom right: $F_1(C) \approx 0$, $F_2(C) \approx 0$, Fitting ≈ 0.

Chan and Vese propose a method to minimize this fitting term and add some regularization terms such as the length of the curve or the area inside the curve. This leads to the energy functional in (2.5), where $\mu \geqslant 0$, $\nu \geqslant 0$, λ_1, $\lambda_2 > 0$ are fixed parameters:

$$\begin{aligned} F(c_1, c_2, C) &= \mu \cdot Length(C) + \nu \cdot Area(inside(C)) \\ &\quad + \lambda_1 \int_{inside(C)} (u_0(x,y) - c_1)^2 dxdy \\ &\quad + \lambda_2 \int_{outside(C)} (u_0(x,y) - c_2)^2 dxdy. \end{aligned} \tag{2.5}$$

In most of their numerical simulations Chan and Vese fix $\lambda_1 = \lambda_2 = 1$ and $\nu = 0$. Thus they consider the minimization problem:

$$\inf_{c_1,c_2,C} F(c_1, c_2, C). \tag{2.6}$$

The level set of the function can be seen in Figure 2.4. The boundary represents the function $\phi = 0$ in the domain Ω.

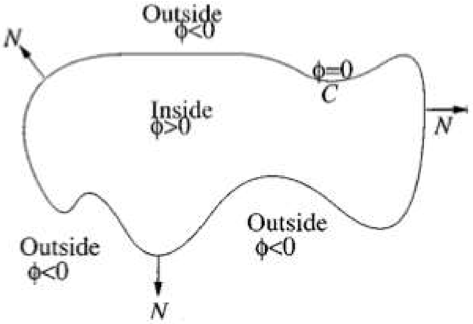

Figure 2.4: Zero level set of a function ϕ and the values of ϕ inside and outside the level set curve, [4]

In order to deduce the Euler-Lagrange formula for the numerical approximations a regularized version the heaviside function,H_ϵ, and dirac delta, δ_ϵ, are used. The Euler-Lagrange equation for ϕ is given by (2.7). The descent direction is parameterized by an artificial time, $t > 0$. The value $\vec{n}$ is the normal derivative of ϕ at the boundary $\partial\Omega$ and $\frac{\partial\phi}{\partial\vec{n}}$ is the normal derivative of ϕ at the boundary.

$$\frac{\partial\phi}{\partial t} = \delta_\epsilon(\phi)\left[\mu \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu - \lambda_1(u_0 - c_1)^2 + \lambda_2(u_0 - c_2)^2\right] = 0, \tag{2.7}$$

$$\phi(0, x, y) = \phi_0(x, y) \text{ in } \Omega, \tag{2.8}$$

$$\frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\overrightarrow{n}} = 0 \text{ on } \partial\Omega. \tag{2.9}$$

One regularization of H is:

$$H_\epsilon = \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{z}{\epsilon}\right)\right). \tag{2.10}$$

Some results from [4] are shown in Figure 2.5 which shows that the method is successfully able to find non-convex shapes in a noisy image.

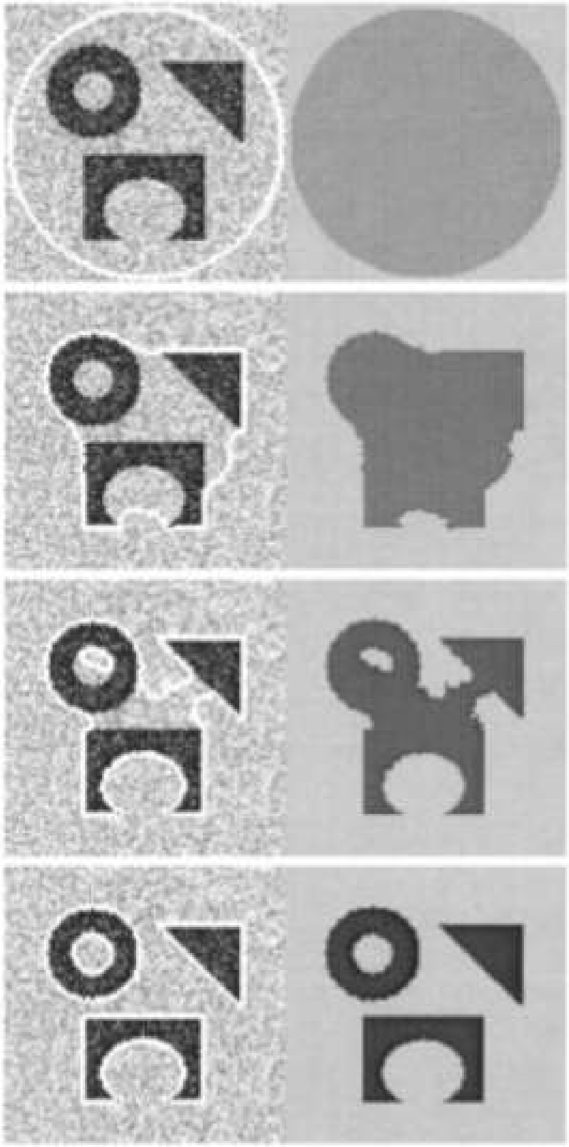

Figure 2.5: Results for [4] on a noisy, non-convex image

2.1.2 A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model

Chan and Vese introduce another method, [17], for image segmentation that extends [4] to segment images that have more than two regions, the goal of this paper is to include images that contain more than two segments and images with complex topologies such as triple junctions. This method also removes other problems such as vacuum and overlap since it naturally produces a set of disjoint regions that span all of Ω. Using the proposed model only $\log_2 n$ functions are needed to represent n image segments.

First consider $m = \log_2 n$ level set functions $\phi_i : \Omega \to \mathbb{R}$. The union of the zero level sets of the functions ϕ_i, $i = 1, ..., m$ will form the edges for the image segments. the authors define the following equations:

$$\begin{aligned} \Phi &= (\phi_1, ..., \phi_m), && (2.11) \\ H(\Phi) &= (H(\phi_1), ..., H(\phi_m)). && (2.12) \end{aligned}$$

$H(\phi)$ may take on values of only 0 or 1. The segments of the domain Ω can now be defined in the following way: two pixels (x_1, y_1) and (x_2, y_2) belong to the same segment if and only if $H(\Phi(x_1, y_1)) = H(\Phi(x_2, y_2))$. Since there are 2^m possibilities for the value of $H(\Phi)$ so in this way up to $2^m = n$ segments in the image are defined.

Some notation to be introduced. The classes are labeled by $1 \leqslant I \leqslant 2^m = n$. A constant vector of averages $c = (c_1, ..., c_n)$, where $c_I = mean(u_0)$ in class I. The characteristic function is defined as:

$$\chi_I = \begin{cases} 1 & \text{if } \phi_I > 0 \\ 0 & \text{if } \phi_I \leq 0. \end{cases} \tag{2.13}$$

This allows the reduced Mumford-Shah energy to be written as:

$$F_n^{MS}(c, \phi) = \sum_{1 \leqslant I \leqslant n = 2^m} \int_\Omega (u_0(x, y) - c_I)^2 \chi_I dxdy + \nu \frac{1}{2} \sum_{1 \leqslant I \leqslant n = 2^m} \int_\Omega |\chi_I dxdy|. \tag{2.14}$$

by simplifying the length term $\sum_{1 \leqslant I \leqslant n = 2^m} \int_\Omega |\chi_I|$ to $\sum_{1 \leqslant i \leqslant m} \nu \int_\Omega |\nabla H(\phi_i)|$ (the sum of the length of the level sets of ϕ_i). This causes some parts of the curve to be counted more than once, thus giving some boundaries a higher weight than others. The authors found that this simplification still produced satisfactory results. This gives the energy to be minimized as:

$$F_n(c, \phi) = \sum_{1 \leqslant I \leqslant n = 2^m} \int_\Omega (u_0 - c_I)^2 \chi_I dxdy + \sum_{1 \leqslant i \leqslant m} \nu \int_\Omega |\nabla H(\phi_i)|. \tag{2.15}$$

Figure 2.6 Shows how $\log_2 n$ level set functions can be used to form n segments. The energy functional from (2.15) is shown in (2.16) for the case where $n = 4$,

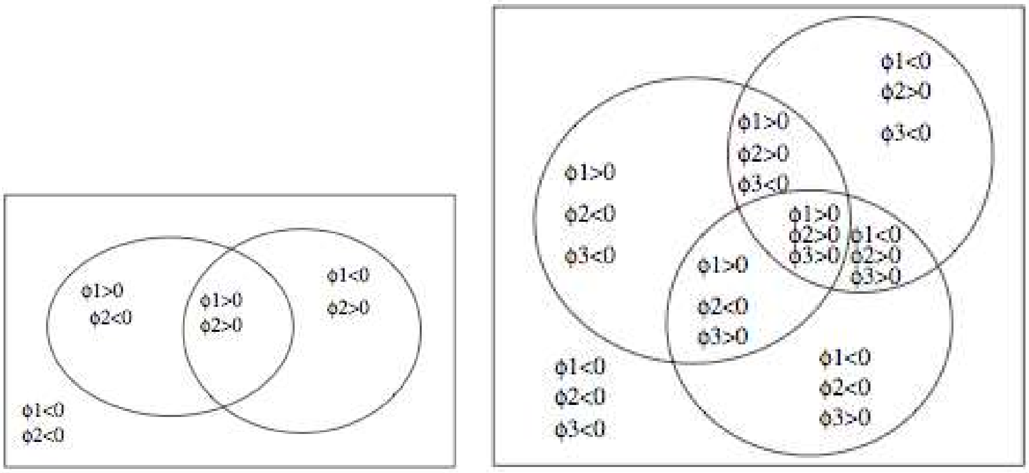

Figure 2.6: Examples of 4 or 8 regions using 2 or 3 level set functions, [17]

which can also be seen in Figure 2.6:

$$
\begin{aligned}
F_4(c,\Phi) \quad = \quad & \int_\Omega (u_0 - c_{11})^2 H(\phi_1)H(\phi_2)dxdy + \int_\Omega (u_0 - c_{10})^2 H(\phi_1)(1 - H(\phi_2))dxdy \\
& + \int_\Omega (u_0 - c_{01})^2 (1 - H(\phi_1))H(\phi_2)dxdy \\
& + \int_\Omega (u_0 - c_{00})^2 (1 - H(\phi_1))(1 - H(\phi_2))dxdy \\
& + \nu \int_\Omega |\nabla H(\phi_1)| + \nu \int_\Omega |\nabla H(\phi_2)|. \qquad (2.16)
\end{aligned}
$$

where:

$$
\begin{aligned}
c_{11}(\Phi) &= mean(u_0) \text{ in } \{(x,y)\colon\ \phi_1(t,x,y) > 0,\ \phi_2(t,x,y) > 0\}, \\
c_{10}(\Phi) &= mean(u_0) \text{ in } \{(x,y)\colon\ \phi_1(t,x,y) > 0,\ \phi_2(t,x,y) < 0\}, \\
c_{01}(\Phi) &= mean(u_0) \text{ in } \{(x,y)\colon\ \phi_1(t,x,y) < 0,\ \phi_2(t,x,y) > 0\}, \\
c_{00}(\Phi) &= mean(u_0) \text{ in } \{(x,y)\colon\ \phi_1(t,x,y) < 0,\ \phi_2(t,x,y) < 0\}.
\end{aligned}
$$

and $\Phi = (\phi_1, \phi_2)$. This allows us to express the image as:

$$
\begin{aligned}
u \quad = \quad & c_{11}H(\phi_1)H(\phi_2) + c_{10}H(\phi_1)(1 - H(\phi_2)) \\
& + c_{01}(1 - H(\phi_1))H(\phi_2) + c_{00}(1 - H(\phi_1))(1 - H(\phi_2)). \qquad (2.17)
\end{aligned}
$$

The Euler-Lagrange equations obtained by minimizing (2.16) with respect to c and Φ, embedded in a dynamical scheme, are given $\phi_1(0,x,y) = \phi_{1,0}(x,y)$, $\phi_2(0,x,y) = \phi_{2,0}(x,y)$:

$$\begin{aligned}\frac{\partial \phi_1}{\partial t} &= \delta_\epsilon(\phi_1)\{\nu \mathrm{div}(\frac{\nabla \phi_1}{|\nabla \phi_1|}) - [((u_0 - c_{11})^2 - (u_0 - c_{01})^2)H(\phi_2) \\ &\quad +((u_0 - c_{10})^2 - (u_0 - c_{00})^2)(1 - H(\phi_2))]\}, \qquad (2.18) \\ \frac{\partial \phi_2}{\partial t} &= \delta_\epsilon(\phi_2)\{\nu \mathrm{div}(\frac{\nabla \phi_2}{|\nabla \phi_2|}) - [((u_0 - c_{11})^2 - (u_0 - c_{10})^2)H(\phi_1) \\ &\quad +((u_0 - c_{01})^2 - (u_0 - c_{00})^2)(1 - H(\phi_1))]\}. \qquad (2.19)\end{aligned}$$

In Figure 2.6 it can be seen how the problem extends to partition the domain Ω into eight segments.

2.1.3 A Fully Global Approach to Image Segmentation via Coupled Curve Evolution Equations

In [21], Yezzi et al. develop a region based active contours segmentation technique to separate the values of certain image statistics over a known number or region types. The algorithm tries to pull apart the values of one or more image statistics (mean, variance, texture, etc).

The authors start by considering an image that consists of just two region types, for example an image that consists of a foreground region R with intensity I^r and background region R^c with intensity $I^c \neq I^r$. They wish to determine an evolution that will attract an initial closed curve C toward the boundary, ∂R, of R. Given that the initial curve will likely contain part of R and part of R^c the average intensities, u and v of R and R^c respectively are bounded above and below by I^r and I^c. This means that the distance between u and v can be used to to determine how well the curve have separated the two regions and their is a an upper bound of $|I^r - I^c|$ that is uniquely obtained when $C = \partial R$. A related strategy which assumes no previous knowledge of I^r or I^c is to minimize the following energy:

$$E = -\frac{1}{2}(u - v)^2. \qquad (2.20)$$

It can be shown that the gradients of u and v are:

$$\nabla u = \frac{I - u}{A_u}\vec{N}, \qquad (2.21)$$

$$\nabla v = -\frac{I - v}{A_v}\vec{N}, \qquad (2.22)$$

where A_u is the area of R and A_v is the area of R^c and $\vec{N}$ is the outward normal to ∂R. These equations can be used to compute the gradient flow of E:

$$\frac{dC}{dt} = -\nabla E = (u - v)\left(\frac{I - u}{A_u} + \frac{I - v}{A_v}\right)\vec{N}, \qquad (2.23)$$

which optimally separates the mean intensities inside and outside C.

This model can be generalized by letting u and v be statistics other than the mean of the regions. For example if you have two areas with the same mean, but different variances you could use the following energy functional:

$$E = -\frac{1}{2}(\sigma_u^2 - \sigma_v^2)^2. \tag{2.24}$$

Again gradient of the two statistics is derived as:

$$\nabla \sigma_u^2 = \frac{(I-u)^2 - \sigma_u^2}{A_u}\vec{N}, \tag{2.25}$$

$$\nabla v = -\frac{(I-v)^2 - \sigma_v^2}{A_v}\vec{N}, \tag{2.26}$$

which gives the curve evolution of:

$$\frac{dC}{dt} = (\sigma_u^2 - \sigma_v^2)\left(\frac{(I-u)^2 - \sigma_u^2}{A_u} + \frac{(I-v)^2 - \sigma_v^2}{A_v}\right)\vec{N}. \tag{2.27}$$

A problem that can be encountered in this type of method is that as the curve evolves past the boundary of the object, it is possible for both statistics to move in the same direction, but the distance between them continues to grow. This is not the desirable behaviour and Yezzi et al. propose the following solution.

First define the following inner product:

$$\left\langle \frac{dC}{dt}, \frac{d\vec{C'}}{dt} \right\rangle = \oint_C \left(\frac{dC}{dt} \cdot \frac{d\vec{C'}}{dt} \right) ds. \tag{2.28}$$

The time derivatives of the statistics u and v under a binary flow are as follows :

$$\begin{aligned} u' &= \langle \nabla u, -\nabla E \rangle, & (2.29) \\ v' &= \langle \nabla v, -\nabla E \rangle, & (2.30) \end{aligned}$$

where $-\nabla E$ denotes the gradient descent flow.

The authors note that the statistics u and v move in opposite directions when $u'v' < 0$. If one of the statistics is moving in the wrong direction, the statistic that is moving in the wrong direction is kept fixed by subtracting from the evolution the component along the gradient direction for that statistic, as shown in the equations below that preserve u or v respectively.

$$\begin{aligned} \frac{dC}{dt} &= -\nabla E + \frac{\langle \nabla E, \nabla u \rangle}{\langle \nabla u, \nabla u \rangle}\nabla u, & (2.31) \\ \frac{dC}{dt} &= -\nabla E + \frac{\langle \nabla E, \nabla v \rangle}{\langle \nabla v, \nabla v \rangle}\nabla v. & (2.32) \end{aligned}$$

This allows for an evolution such that the statistics should always move in opposite directions.

2.2 Watershed Method

The watershed method is a segmentation algorithm that is often used on images [14]. It can be thought of as a flooding process. A 2D image can be thought of as a topography with the pixel intensity describing the height. If "water sources" are placed at various locations and then allowed to "flood" the image, the water coming from different water sources will meet along watershed lines. These watershed lines form the boundaries for the different regions.

First the seeds for each of the regions are selected. The seeds do not need to be a single point or a minimum of a particular area. All seeds have a unique identifier. These are the areas where the water originates from in the flooding analogy. The water continues to flood the area, eventually neighbouring regions will join unless a dam (watershed line) is erected between them. For the watershed method the dams are considered to be infinitely tall. When a new region floods it will be given the same label as its neighbouring region that caused the flooding. Eventually the image is divided into non-intersecting regions that are referred to as catchment basin and are bordered by the watershed lines. There are exactly the same number of basis as types of seeds. The algorithm is sensitive to the number of initial seeds as too many will result in an over segmentation of the image, while too few will result in different regions being joined together. The algorithm is also sensitive to where the seeds are placed, as this will determine how the image "floods." Good seed selection is very important to the outcome of the algorithm.

The algorithm is as follows:

1. Select the seeds for each region, place seeds into a FIFO queue according to the value of the seed.

2. Dequeue the next seed.

3. All pixel neighbours are retrieved from the image. All neighbouring pixels without an ID are given the ID of the current seed and placed in the queue according to their pixel value. All neighbouring pixels that already have an ID are ignored.

4. Return to step 2.

2.3 Spectral Graph Partitioning

Spectral graph partitioning is the segmentation of a graph into two pieces based on the the eigenvectors and eigenvalues of the Laplacian matrix for the graph [16], [5].

First some terms are needed to describe graph $G = (V, E, w)$ where G is an edge weighted, undirected graph with no self loops. V is a set of n vertices labeled $1, ..., n$, E is a set of m edges and w is an edge weighting $\in [0, 1]$.

- w_{ij} is the weight between edges i and j
- W, weighted adjacency matrix:

$$W_{ij} = \begin{cases} w_{ij} = w(i,j) & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

- d_i, The weighted degree of a vertex:

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- D, the weighted degree matrix

$$D_{ij} = \begin{cases} D_{ij} = d_i & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

- Generalized Laplacian of G is $L = D - W$
- Normalized Laplacian of G is $\mathcal{L} = D^{-1/2} L D^{-1/2}$

To cut this graph into two pieces A and B, a measure is needed to determine the quality of the cut. A normalized cut is measured as follows, where $assoc(A, V)$ is the sum of the weighted degree of the vertices in A and $cut(A, B)$ is the sum of the weights of the edges joining A and B:

$$NCut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}. \tag{2.33}$$

When a normalized cut is minimized the result is a partition such that there is high connectivity in the two parts, but low connectivity between them. Through relaxation of the optimization problem, the normalized cut can be rewritten as:

$$\min_x \frac{x^T L x}{x^T D x}, \; ||x|| = N. \tag{2.34}$$

Using the weak duality theorem from optimization, we are able to relax this problem to the following generalized eigenvalue problem:

$$Lx = \lambda D x, \tag{2.35}$$

$$(D - W)x = \lambda D x. \tag{2.36}$$

The smallest solution to (2.36) is trivial, $x = 0$, so the second smallest solution is considered. This corresponds to the second smallest eigenvalue/eigenvector pair. The vertices are split along the zero crossing.

To apply this technique to image segmentation a graph must be formed from the image. This can be done using decreasing functions that measure the similarity in pixel value of any two pixels as well as their distance apart. That is each pixel in the image becomes a node in the graph. Two pixels are connected by an edge based on a function such as the one below:

$$w = e^{-\frac{|I(x_1,y_1)-I(x_2,y_2)|_2^2}{\sigma_I}} \cdot e^{-\frac{|(x_1,y_1)-(x_2,y_2)|_2^2}{\sigma_x}}, \tag{2.37}$$

where σ_I and σ_x are constants based on the expected distribution of the pixel intensities and distance of related pixles.

This function decays with distance, however to create a sparse graph an additional constraint of only considering pixels within a certain neighbourhood to have an edge is used. This will remove edges with a very low weight.

The spectral clustering method using the second smallest eigenvalue/eigenvector pair naturally gives a partition into two parts, however if to partition the graph into more pieces other small eigenvectors are considered. The third smallest eigenvalue/eigenvector would split the graph into three pieces, the fourth into four, etc. Another natural extension is to take several small eigenvectors, say n of them and form points in $\mathbb{R}^n$ from the corresponding entries. A clustering method can then be used on these points.

2.4 Clustering - K-Means

The K-Means Clustering algorithm is an algorithm to cluster N points into K distinct groups, S_j, $j = 1, ..., K$. The algorithm is designed to minimize the distance between each point and the geometric centroid of the group to which it belongs

$$\min \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|_2. \tag{2.38}$$

The data point x_n is a vector representing the n^{th} data point, μ_j is a vector representing the geometric centroid of group j and the euclidian distance is used as a measure, although other measures (manhattan distance, L_1 norm, etc) are also possible depending on the application of clustering. While the algorithm cannot be guaranteed to find the global minimum it is commonly used due to its ease of implementation. The algorithm is as follows:

1. Randomly assign each of the N points to a group j.
2. Compute the centre of each group.
3. Assign each point to the centre that it is closest to.
4. Repeat steps 2 and 3 until a stopping criteria has been met (i.e. no points change groups)

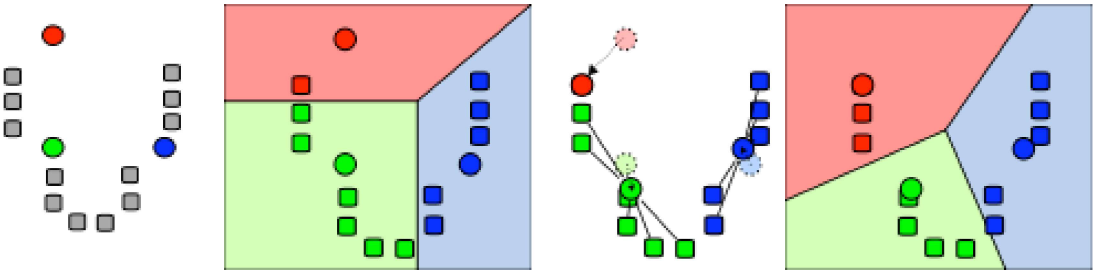

Figure 2.7: Sample iteration of K-Means algorithm.[2]

Figure 2.7 shows an iteration of the algorithm. In the leftmost image the centres are assigned, in the second image the points are assigned to their closest centres, the next image shows the recalculation of the centres based on the new groups and finally the last image shows the partition.

Since the algorithm cannot guarantee global convergence, but is extremely fast to run in practice, it is common to run the algorithm several times with different initial centres and then choose the result that resulted in the lowest cost. The number of groups K is taken by the algorithm as an input parameter. This contributes to the speed of the algorithm, but is also a drawback of the algorithm since the wrong K can cause poor clustering results. Other drawbacks include that all of the attributes are given the same weight. The algorithm is sensitive to its initial starting conditions, especially if there are only a few points. Since a distance metric is used for clustering, the result is circular clusters. Finally the algorithm is sensitive to outliers since it relies on a mean of the group.

There are suggested solutions to many of these drawbacks. Random starts reduce chance of finding only a local minimum and median can be used instead of mean to reduce the effect of outliers.

The applications of K-means include the unsupervised learning of neural networks, image processing, computer vision, artificial intelligence, classification and more.

[2] Source: Wikimedia Commons, Wenston Pace, 2007. Distributed under GNU Free Documentation License

Chapter 3

Literature Review

Most of the current research into the problem of segmenting images of cells is focused on other types of cell images such as fluorescent, phase contrast or confocal images. This is because bright-field images show the least amount of detail and are the hardest to segment.

3.1 Fluorescent Images

The segmentation of fluorescent images is much simpler than the segmentation of bright-field images because the fluorescent images usually show only the nuclei of the cells, which are small, convex, bright and do not overlap.

Dzyubachyk et al. [7] present a method to segment and track multiple cells in fluorescent microscopy images in their paper . The framework they present is based primarily on a level set segmentation, but also makes use of some other segmentation techniques. Each object (cell or nucleus) is represented by a level set function ϕ_i,$i = 1, \ldots, n$ where n is the number of objects in the 3D image. The final segmentation for an image is found by minimizing an energy functional which is composed of both imaged based and smoothness components. A probability based approach is used in this paper, where the probability that a pixel belongs to a particular region is considered in the functional. It is also assumed that all the regions have Gaussian distributions over the pixel intensities which allows for the descriptions of the internal energies of both the nuclei and the background shown in (3.1) where Ω_i is region inside the the i^{th} level set function:

$$e_i = \log p(I \mid \Omega_i) = \log \sigma_i^2 + \frac{(I - u_i)^2}{\sigma_i^2}. \tag{3.1}$$

Dzyubachyk et al. use some heuristic approaches to improve their algorithm. A single level set function is evolved on the first image of a sequence to find connected components. A watershed algorithm is then used to separate any cells that are lumped together. The stopping criterion of the level set algorithm is a sign change based criterion, which considers the number of voxels that change sign. A radon transform based cell separation was used to attempt to prevent the level set function of one cell to capture part of another cell. Finally the authors run another segmentation method on the now segmented image to quickly determine if they believe that any new cells have entered the field of view. The preliminary results that are presented in [7] show a good segmentation, however the authors do not discuss how to deal with mitosis which is essential when dealing with sequences of bright-field cell images. Furthermore a single level set function does not work on bright-field cell images, which is discussed in chapter 4.

Wang et al. [18] present another approach for segmenting fluorescent images in their paper. Their approach uses both the intensity and information about the shape of the cell to segment the image. A binarization is used to segment clusters of cell from the background. The algorithm used a cubic B-spline to estimate the background. If obvious differences exist then the pixel is assigned to be cell, otherwise it is background. The next step is to find the local maxima. A new image is generated by adding the original image and the image with a distance transform performed on it. The resultant image has its noise suppressed and local maxima where there are likely to be cell centres. Finally the local maxima are detected using thresholding techniques and the watershed algorithm is used with the local maxima as the seeds. This technique would not work well on bright-field images since clusters of cells cannot be identified with thresholding.

3.2 Phase Contrast Images

Phase Contrast images are images that are taken to make highly transparent objects more visible, it is usually used to view intercellular structures [9].

Debeir et al. [6] present a solution in their paper to segment phase contrast images. A segmentation like the watershed algorithm is used, but used random markers and perturbations to create several possible solutions that can be combined to form a better solution. Random noise is added to the markers that are found for the watershed method. This causes there to be slightly different basins each time the algorithm is initialized. The image is also perturbed by adding a constant slope to the image. The value of the slope is set arbitrarily. The slope results in the different weak gradients being reinforced depending on the slope value. The segmentation is run several times on the same image with different markers and slopes, and then a counter for each pixel which counts the number of times that the pixel is part of the region is used. Pixels that are often counted as cell have a high count, which allows the cells to be found. Since the main component of this algorithm is the watershed algorithm, it does not seem to be able to track different cells from frame to frame which is needed for the bright-field segmentation solution.

3.3 Confocal Images

Confocal images are images from light microscopes that allow the user to focus and get crisp images at different specimen depths [9].

Beaver et al. [3] present a solution for segmenting nuclei using a performance based thresholding technique in their paper. The segmentation is based on the finding a threshold for each nuclei for each frame in the stack. The problem is that the threshold changes from frame to frame and from nuclei to nuclei. Since the authors are not looking for a global threshold value the problem becomes one of trying to choose the best threshold for each nuclei. A scoring function is used to determine the quality of each threshold. This function used features of the segmented area such as the mean of the two axes, the ratio between them, the compactness, effective diameter, the convexity, the eccentricity and the circularity, though the authors point out that this is a subset of possible features that could be used. A machine learning algorithm is used to to learn how to classify the regions produced by a given threshold into good, too small or too large. The authors chose to use adaBoost, a supervised learning algorithm, which has many benefits such as not over-fitting the data, and it can be improved without retraining as the training set grows. The method did very well when compared to other thresholding methods (96% compared to 80% for Otsu).

A technique developed by Marcuzzo et al. [11] is used to segment the cells in the root of a plant. It consists of three steps, image registration and filtering, watershed segmentation and finally a classification step. The step of image registration is necessary due to the fact that the root changes direction as it grows, so all images are registered so that the root has the same orientation, this will allow the cells to be more easily compared. The image is then filtered to removed the noise that is often present in confocal images. The watershed transform is applied to the filtered image. Watershed transforms often lead to over segmentation and the images of the arabidopsis root are no exception. The solution the authors employ is to prune badly segmented regions after the segmentation step. A discrete cosine transform is used to describe each region, based on this description a SVM (Support Vector Machine) cell classification is used to prune the non cell regions.

These algorithms are also not practical for the segmentation of bright-field cell images because they do not provide a way to track specific cells.

3.4 Bright-field Images

Finally the segmentation solutions to bright-field images are considered.

Korzynska et al. [10] present a semi-automatic solution for the segmentation of bright-field cell images in their paper. The algorithm exploits three known features of bright-field cell images. The first feature is that the texture of the cell is different from that of the background. The second feature is that for most parts of the cell the boundary gradient is well defined. Finally consecutive images in a time series are very similar. The algorithm requires operator input to initialize the segmentation for each cell and to set several parameters. The operator is also able to go back and correct segmentations by adjusting parameters. The algorithm seems to be only defined for cells that are well separated and that are fully inside the field of view. The algorithm also requires a lot of human interaction.

Rehan et al. [2] present a method for segmenting cells in bright-field images. This method of segmenting images requires that a derivative image is created across the focal plane. That is an image is acquired below and another from above the focal plane. The derivative image is the difference of the two images. A local phase map that is designed to detect the presence of features is then created using a mongenic signal with a quadrature triple filter. Unwanted features are removed by masking them with a threshold variance map of the derivative image. This new corrected local phase map is thresholded to divide the image into background and cell regions. A manual step is now required to split touching cells, the user must click once inside each cell. The 'clicks' become the seeds for a region growing algorithm. A level set method is used next to ensure that all cells are correctly segmented from their neighbours. This method was able to correctly identify 83% ± 7% of all cell pixels, which improved on the authors previous work [1]. This method is not practical for all applications, space wise, since at least 3 times as many images are required and it also requires the images to be taken in a very specific way.

Wu et al. [19] present an early solution to the segmentation of unstained living cells in their paper. The goal of [19] paper is to present a solution to segment living cells that are a suspended in a 3D gel solution. The segmentation solution works only on cells that are contained in their own window with no other cells entering or leaving. The authors present a two stage segmentation in which they first find an approximate region that the cell resides in by segmenting the cell and near background from the distant background. The second step is to segment the cell from the remaining background in the approximate region. The first segmentation is completed by computing the variance at each pixel (determined by the variance of a window centered at the given pixel) and then a minimum-error thresholding method is used to find the approximate region on the "variance" image. The second step is completed using Otsu thresholding. The authors found their results to be satisfactory, but the algorithm does not work for multiple cells in an image, furthermore even with the cell taking up most of the image frame, good results with thresholding techniques have not been produced for the data set being investigated in this paper.

Chapter 4

Algorithms

Several algorithms, based on the background information found in chapter 2, have been implemented to solve the bright-field cell image segmentation problem. The details are in the following sections.

4.1 Multiphase and Watershed

The first method considered is a combination of the multiphase active contour [17] which was discussed in section 2.1.2 and the watershed method [14], [15] discussed in Section 2.2.

The results from a single level set function or even the multiphase level set algorithm on a cell image is not enough to use to segment the cells. This was expected based on the preliminary thresholding attempts (shown in Section 6.1). The segments that result from using a level set method with three level set functions are shown in Figure 4.1. Much of the interior of the cell is still contained within the same region as the background (see Figure 4.1, second row, last column), but those regions are disconnected from the region that actually represents the background. However the level set method was able to produce large regions in the cell interior to use as markers for the watershed method. The three phase method still places some crucial regions together, but this is because the intensities are so similar. It is for this reason that it is not attempted to obtain better results with four level set functions (i.e. sixteen regions). The information simply is not in the intensity of the image.

These results are instead used for the the seed for the watershed algorithm, it can be seen in the sequence of images that the one on the far left contains the halo for the cell (the brightest region), the other regions contain mostly small dark areas on the cell interior, the final segment contains the background and cell interior. Through heuristic approaches the background component is separated from the cell component and use the cell component along with the other darker regions to

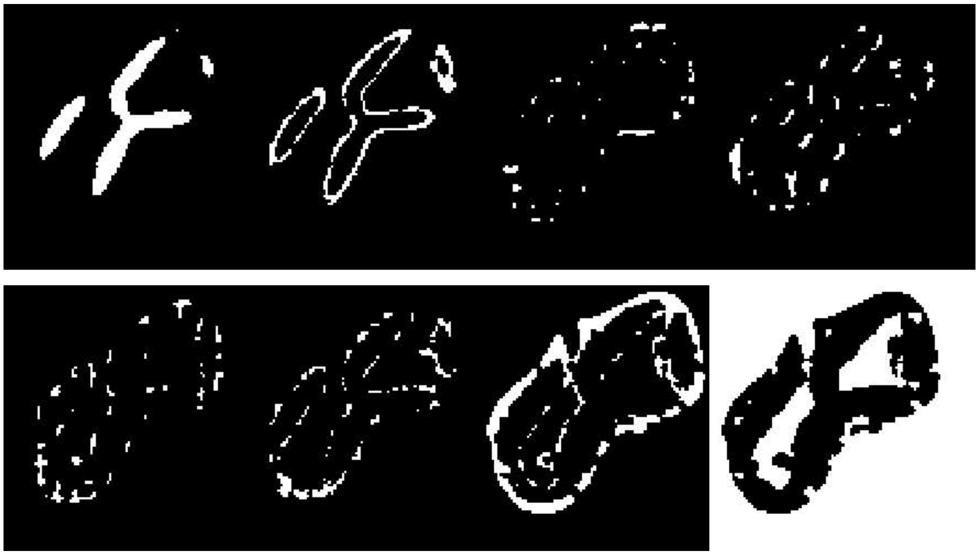

Figure 4.1: Regions from multiphase segmentation

form the seed for watershed. By having a variety of intensities in the watershed seed, the algorithm is able to capture the cell boundary well.

Two methods of extracting seeds for the watershed method were implemented. The first method is based on using the multiphase level set method to extract regions that are interior points of the cell. The image set that was used for these results has the interior of the cells darker than the background and the halo. For this reason the two regions corresponding to the darkest average intensities were selected to make the seed. It is common for the background to be included in these two regions, but in this case there are lighter regions, separating the cell region from the background region (see Figure 4.1). The binary image formed from choosing the two regions that contain the darkest regions is then eroded using morphological operators to take care of the cases where the cell and background are connected by small threads. This also pushes the seed back from the boundaries of the region in case the region contained some background. Because the background is separate from the cell, the algorithm simply removes the pixels from the seed that correspond to background. This binary image is used to select the pixels that form the seeds for the interior region for the watershed algorithm from the original image. The boundary of the image used for segmentation is used as the seeds or the outer region of the watershed segmentation.

The second method used the results from the multiphase level set segmentation as well. This time the method is based on the idea of segmenting the background rather than the cell. Since the multiphase method was used a square region of the image containing the cell, it is known that the the region containing the background will contain the most non zero elements. The algorithm selects this region, and then inverts the binary image, filling any holes in the resulting image. This area is now an approximation of where the cell is located. It is eroded to be used as a seed for the watershed algorithm. The seed for the outside area of the watershed is chosen to be the same as in the first method.

The watershed algorithm results in a binary image with two regions, cell and background. A simple level set method is used on this image to extract the the boundary of the cell. This improves the smoothness of the boundary as the watershed method often produces a saw-tooth pattern. Furthermore, the level set method does a better job of detecting cell division, as it is able to ignore thin lines of pixels connecting two segments.

4.2 Statistical Level Set Approach

The second approach to segmenting the cell images comes from the papers by Yezzi et al. [21], [20]. Three methods were tested from this paper, a method using the mean for the statistics, a method using the variance for the statistics and finally a method that used both the mean and the variance. All of the variations of this method were implemented with an up-winding, finite difference method to solve the level set problem.

4.2.1 Segmentation based on Mean

Recall that the goal of Yezzi's method using the means of the two regions to be segmented was to minimize the following energy:

$$E = -\frac{1}{2}(u-v)^2, \tag{4.1}$$

where u is the mean of the foreground of the image and v is the mean of the background of the image. A level set method was used to accomplish this segmentation with the speed term F being:

$$F = \nabla E = -(u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right). \tag{4.2}$$

Additional constraint were derived to force the statistics to move in opposite directions using the equations below:

$$\begin{aligned} \frac{dC}{dt} &= \frac{u-v}{A_v}((I-u) + \gamma_u(I-v),) & (4.3) \\ \frac{dC}{dt} &= \frac{u-v}{A_u}((I-v) + \gamma_u(I-u)), & (4.4) \end{aligned}$$

where (4.3) is used to prevent the statistic u from moving in the wrong direction and (4.4) is used to prevent v from moving in the wrong direction. A statistic moves in the wrong direction if both statistics move in the same direction.

4.2.2 Segmentation based on Variance

To segment based on variance the following energy equation is minimized:

$$E = -\frac{1}{2}\left(\sigma_u^2 - \sigma_v^2\right)^2, \tag{4.5}$$

where σ_u^2 is the variance of the foreground of the image and σ_v^2 is the variance of the background of the image. A level set technique is used to accomplish this segmentation with the speed term F being:

$$F = \nabla E = -(\sigma_u^2 - \sigma_v^2)\left(\frac{(I-u)^2 - \sigma_u^2}{A_u} + \frac{(I-v)^2 - \sigma_v^2}{A_v}\right). \tag{4.6}$$

Additional constraints were derived to force the statistics to move in opposite directions using the equations below:

$$\begin{aligned}\frac{dC}{dt} &= \frac{(\sigma_u^2 - \sigma_v^2)}{A_v}((I_u)^2 - \sigma_u^2) + \gamma_u((I-v)^2 - \sigma_v^2), && (4.7)\\ \frac{dC}{dt} &= \frac{(\sigma_u^2 - \sigma_v^2)}{A_u}((I_v)^2 - \sigma_v^2) + \gamma_u((I-u)^2 - \sigma_u^2), && (4.8)\end{aligned}$$

where (4.7) is used to prevent the statistic σ_u^2 from moving in the wrong direction and (4.8) is used to prevent σ_v^2 from moving in the wrong direction.

4.2.3 Segmentation based on Mean and Variance

Following the work by Yezzi et al. [21], a combined energy function using both mean and variance was derived:

$$\begin{aligned}E &= -\lambda_1\frac{1}{2}(u-v)^2\\ &\quad -\lambda_2\left(\frac{1}{2}(\sigma_u^2 - \sigma_v^2)^2 - (\sigma_u^2 - \sigma_v^2)\left(\frac{(I-u)^2 - \sigma_u^2}{A_u} + \frac{(I-v)^2 - \sigma_v^2}{A_v}\right)\right)\end{aligned} \tag{4.9}$$

where u is the mean of the foreground of the image, v is the mean of the background of the image, σ_u^2 is the variance of the foreground of the image and σ_v^2 is the variance of the background of the image. The constants λ_1 and λ_2 are used to weight the importance of the energy of the combined statistics. A level set technique is used to accomplish this segmentation with the speed term F being:

$$\begin{aligned}F = \nabla E &= -\lambda_1(u-v)\left(\frac{I-u}{A_u} + \frac{I-v}{A_v}\right)\\ &\quad -\lambda_2(\sigma_u^2 - \sigma_v^2)\left(\frac{(I-u)^2 - \sigma_u^2}{A_u} + \frac{(I-v)^2 - \sigma_v^2}{A_v}\right).\end{aligned} \tag{4.10}$$

Table 4.1: Statistic Movement

u	v	σ_u^2	σ_v^2	Equation
True	True	True	True	4.10
False	True	True	True	4.11
True	False	True	True	4.12
True	True	False	True	4.13
True	True	True	False	4.14
False	True	False	True	4.15
False	True	True	False	4.16
True	False	False	True	4.17
True	False	True	False	4.18

Additional constraints were derived to force the statistics to move in opposite directions using the equations below:

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla u \rangle}{\langle \nabla u, \nabla u \rangle} \nabla u, \tag{4.11}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla v \rangle}{\langle \nabla v, \nabla v \rangle} \nabla v, \tag{4.12}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla \sigma_u^2 \rangle}{\langle \nabla \sigma_u^2, \nabla \sigma_u^2 \rangle} \nabla \sigma_u^2, \tag{4.13}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla \sigma_v^2 \rangle}{\langle \nabla \sigma_v^2, \nabla \sigma_v^2 \rangle} \nabla \sigma_v^2, \tag{4.14}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla u \rangle}{\langle \nabla u, \nabla u \rangle} \nabla u + \frac{\langle \nabla E, \nabla \sigma_u^2 \rangle}{\langle \nabla \sigma_u^2, \nabla \sigma_u^2 \rangle} \nabla \sigma_u^2, \tag{4.15}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla u \rangle}{\langle \nabla u, \nabla u \rangle} \nabla u + \frac{\langle \nabla E, \nabla \sigma_v^2 \rangle}{\langle \nabla \sigma_v^2, \nabla \sigma_v^2 \rangle} \nabla \sigma_v^2, \tag{4.16}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla v \rangle}{\langle \nabla v, \nabla v \rangle} \nabla v + \frac{\langle \nabla E, \nabla \sigma_u^2 \rangle}{\langle \nabla \sigma_u^2, \nabla \sigma_u^2 \rangle} \nabla \sigma_u^2, \tag{4.17}$$

$$\frac{dC}{dt} = -\nabla E + \frac{\langle \nabla E, \nabla v \rangle}{\langle \nabla v, \nabla v \rangle} \nabla v + \frac{\langle \nabla E, \nabla \sigma_v^2 \rangle}{\langle \nabla \sigma_v^2, \nabla \sigma_v^2 \rangle} \nabla \sigma_v^2. \tag{4.18}$$

It is more complicated in this case to determine which statistic is moving in the wrong direction since there are more than two possibilities as shown in Table 4.1. True indicates that a statistic moved in the proper direction, while false indicates that it did not and the equation number describes the equation that represents the speed term in that case. For example, if u and σ_u both moved in the incorrect direction, but the other two directions moved correctly, then (4.15) is used.

4.3 Spectral-K-Means Segmentation

The final approach that will be discussed in this paper is an approach based on the spectral graph theory [5], [16].

First of all the Laplacian graph representation of the image is found in the following way. A graph, G is created of the image, the weighted adjacency matrix, W and the weighted degree matrix, D as described in Section 2.3 are found next. The normalized laplacian $\mathcal{L}$ is then computed as well as the m eigenvectors corresponding to the m smallest eigenvalues of $\mathcal{L}$. Ignoring the eigenvector corresponding to the smallest eigenvalue since it does not provide any information, a point in $\mathbb{R}^{m-1}$ is formed for each pixel that contains its entry from the other $m-1$ eigenvalues. Next K-Means is run using p random starts on this data set. This will divide the image into k regions. For the experiments in the following section it was found that $m = 10$, $k = 9$ and $p = 10$ provided good results in most cases.

The second part of the algorithm is used to determine the approximate area where the cell is located. The goal is to maximize the difference of the variance between the two groups of pixels, thus the algorithm starts with an initial partition and then looks at all the pixels on the inside and outside boundary of this partition. If the difference in variance between the two regions increases by having the pixel change groups, then the pixel is placed in the other region. This continues like this until no more pixels change regions. The restriction that the statistics must move in opposite directions is also imposed. Thus, even if changing a pixel increases the difference between the variances of the two groups, the pixel is not moved if both variances increased or decreased.

The segments from the K-Means partitioning that are contained nearly entirely within the cell area found by the second part of the algorithm are selected. Theses segments are then combined as the area where the cell is located. A summary of the algorithm is below

- Form $\mathcal{L}$ based on the image
- Solve for the m eigenvectors that correspond to the m smallest eigenvalues
- Disregarding the smallest eigenvector, form points in $\mathbb{R}^{m-1}$ for each of the pixels in the image.
- Use K-Means to find k regions of these points.
- Find Approximate location of cell
- Locate all regions from K-Means segmentation found primarily within this area

Unlike the other algorithms presented in this section, the spectral K-Means segmentation is not a deterministic algorithm. The segmentation depends on the placement of initial centres. In practice, by running the K-Means algorithm several times with random starts, the same segmentation is normally seen each time the segmentation algorithm is executed.

Chapter 5

Multi-cell Segmentation

In order for the segmentation to be useful we need to be able to segment images that contain more than one cell. The following algorithm shows how we accomplish this.

Each cell in the image is segmented individually so that they can individually be tracked through a series of images. This will allow the program to keep track of cells as the move, grow, divide and die.

In order to segment each cell individually we will use the location of the cell that was found in the previous frame. We make an assumption here that cells do not move very far in between frames. This assumption is not unreasonable because it is possible to increase the rate at which images are taken. Also if the cell moves too far in between frames even a human will not be able to be sure which cells are which. All other cells except the one being segmented will be grayed out to match the background intensity by graying out an area including and around the last know location of the cell.

We now select the portion of the image where we expect to find the cell. By using the cell's previous location and size we select a window that includes the probable area that contains the cell. We then segment that area to find the cell using any of the techniques in the previous chapter.

We repeat this for each cell in the image. This method requires that a technician manually locate approximate location of each cell in the first image so that the algorithm has a way to segment one cell at a time.

A possible extension to allow this to work for cells that are very close together or touching is to repeat the segmentation on a single frame several times, each time (other than the first which is the same as above), the location of the cell found in the previous segmentation attempt is used to hide the location of the cell in the next iteration. In this way if the estimate of one cell's location blocks out another cell, a tighter and tighter estimate can be used to hide the location of the cells.

Chapter 6

Results

The following section presents the results of the algorithms discussed in chapter 4 and chapter 5. All of the following cell images are bright-field cell images from the C2C12 data set from the Ontario Institute for Cancer Research and the Department of Medicine and Human Genetics at McGill University.

6.1 Multiphase Watershed Combination

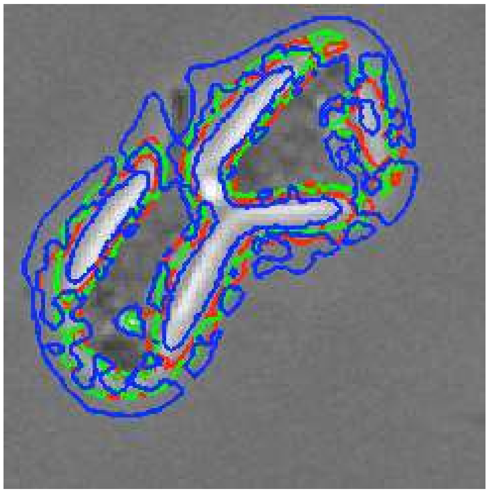

Figure 6.1: Multiphase Segmentation results of a cell nearing division with both weak boundaries and a broken halo.

Figure 6.1 shows an example of the result of Chan and Vese's multiphase level set segmentation algorithm, [17], on a bright-field cell. Three level set functions were used for a total of eight regions. Figure 4.1 in Chapter 4 shows the individual regions. It is clear from this image that the cell interior and background are part of the same region. Many of the other regions are very small and similar in intensity so it is unlikely that a fourth level set function (sixteen regions) will produce better results.

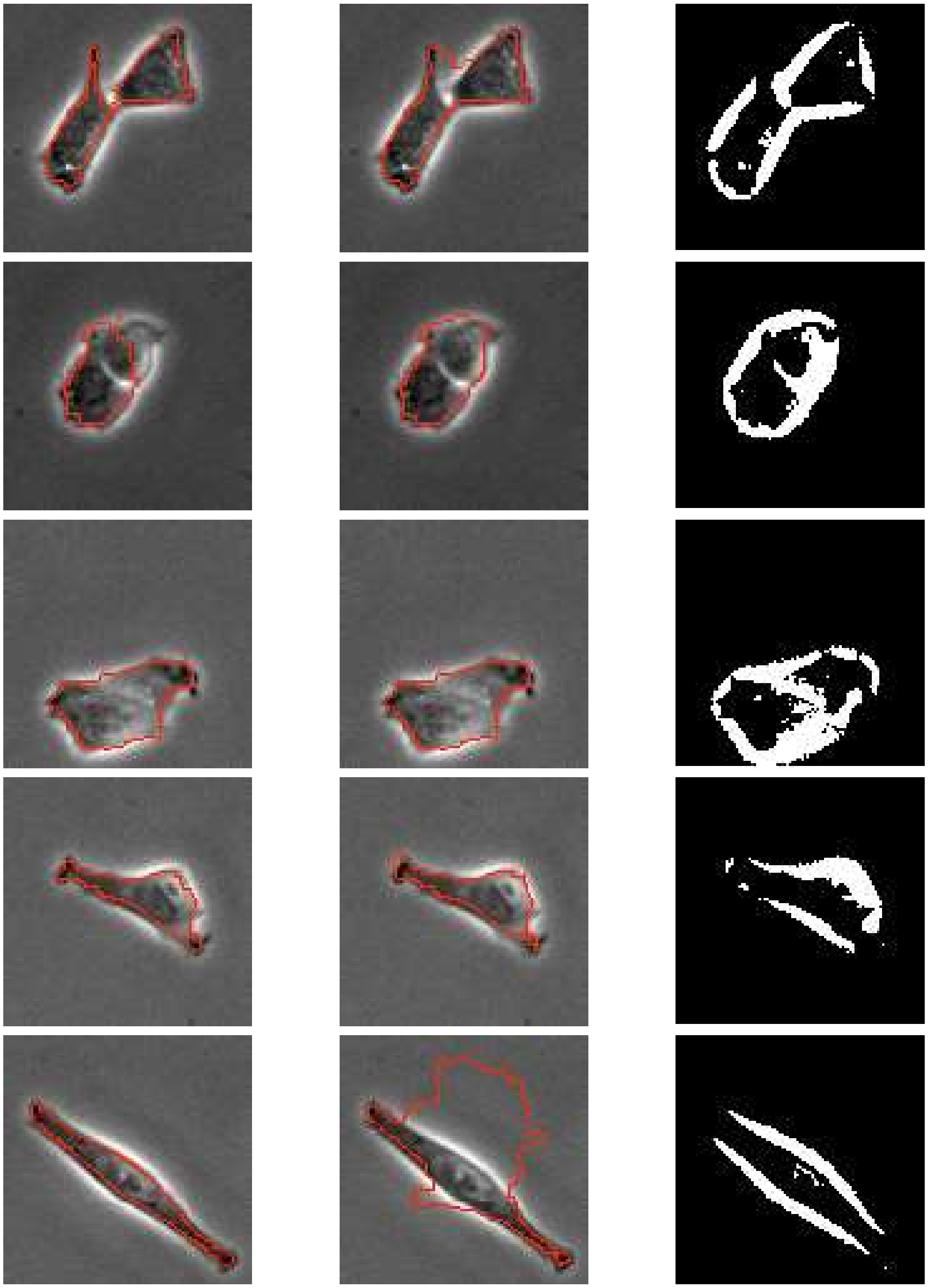

Figure 6.2: Results of Multiphase Watershed Algorithm. Left column used method one for seed extraction, while the middle column used method two. Right column shows the results of an Otsu Threshold algorithm.

Figure 6.2 shows the results for the Multiphase Watershed combination algorithm, using two different seed choices for the watershed algorithm based on the regions identified from the multiphase step. The left most column of the figure shows the results when using the two darkest regions as the seed and the middle column shows the results of using the skeleton of the non background region as the seed (algorithms described in more detail in Section 4.1). The final column is the result of 'graythresh' in MATLAB, which uses Otsu's algorithm. It is presented here to demonstrate that the background and cell are approximately the same intensity, and that simple thresholding methods do not work with this type of image.

We see from the results that method one gives a more conservative estimation of the location of the cell than does method two. While the first method is more likely to miss portions of the cell (row 2 of Figure 6.2), method two more often includes the background (rows 1 and 5 of Figure 6.2). It is likely that the part of the cell has been missed in row 2 column 1 because it is much lighter than the remainder of the cell. If a portion of the light part of the cell was not contained in the watershed seed, the watershed algorithm will not know to add it to the cell region as it looks different from the rest of the cell region. For the second method the background is sometimes included in the cell region because the method for extracting the seed makes it more likely that a piece of the background is included in the seed for the cell. If this is the case the cell region will quickly begin to flood the background area.

6.2 Statistical Level Set Approach

Figure 6.3 show some results from the Statistical Level set method, using the mean as the the statistics to separate the regions. As we can see from the results, this algorithm does not do a good job of locating the cell boundary, rather it tends to find a large area surrounding the cell, but with the cell interior as part of the background region. As we know from the thresholding example, the interior of the cell and the background are about the same intensity. For this reason it is not reasonable to expect that a method based on the mean intensity of a region be able to segment the image into two regions, cell and background.

Figure 6.4 is the segmentation results of the statistical level set method using a weighted combination of the mean and the variance of the regions. This algorithm also is unable to locate the cell boundaries. It instead seems to focus on the halo. The halo region is different from the rest of the image in intensity and in variance.

The method using just variance to segment the images did not work at all, and all segmentation attempts seem to reduce down to a single point. A single point has no variance, while the the rest of the image would have a much higher variance.

6.3 Spectral-K-Means Segmentation

Figures 6.5, 6.6, 6.7, 6.8 show the results of the Spectral-K-Means Segmentation on a set of cell images. The algorithm is described in detail in section 4.3. The first

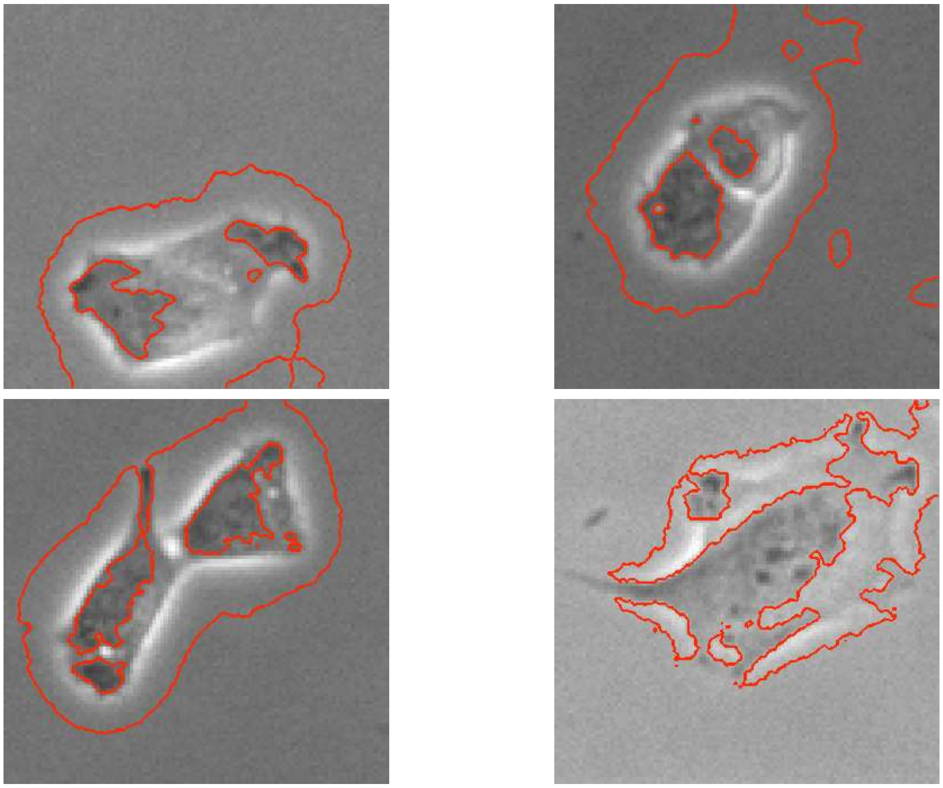

Figure 6.3: Results of the Mean Level Set Method

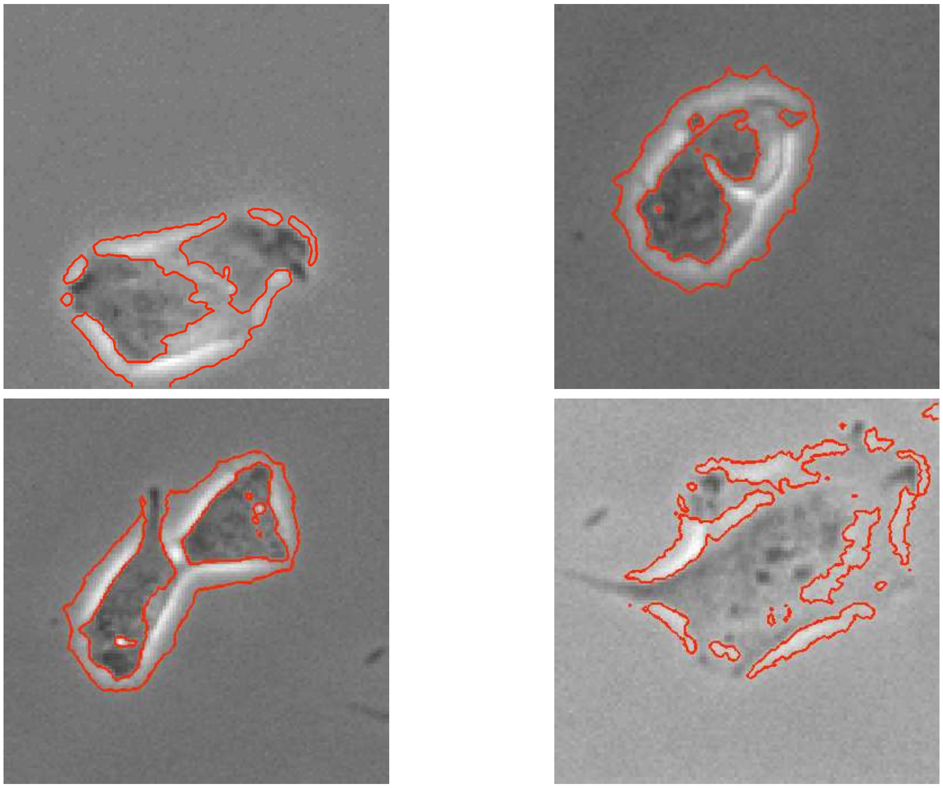

Figure 6.4: Results of the Mean and Variance Level Set Method

column shows the results of the K-Means algorithm on the eigenvectors. The second column is the result of the simple method to maximize the difference in variance and finally the third column shows the final segmentation result. As we can see

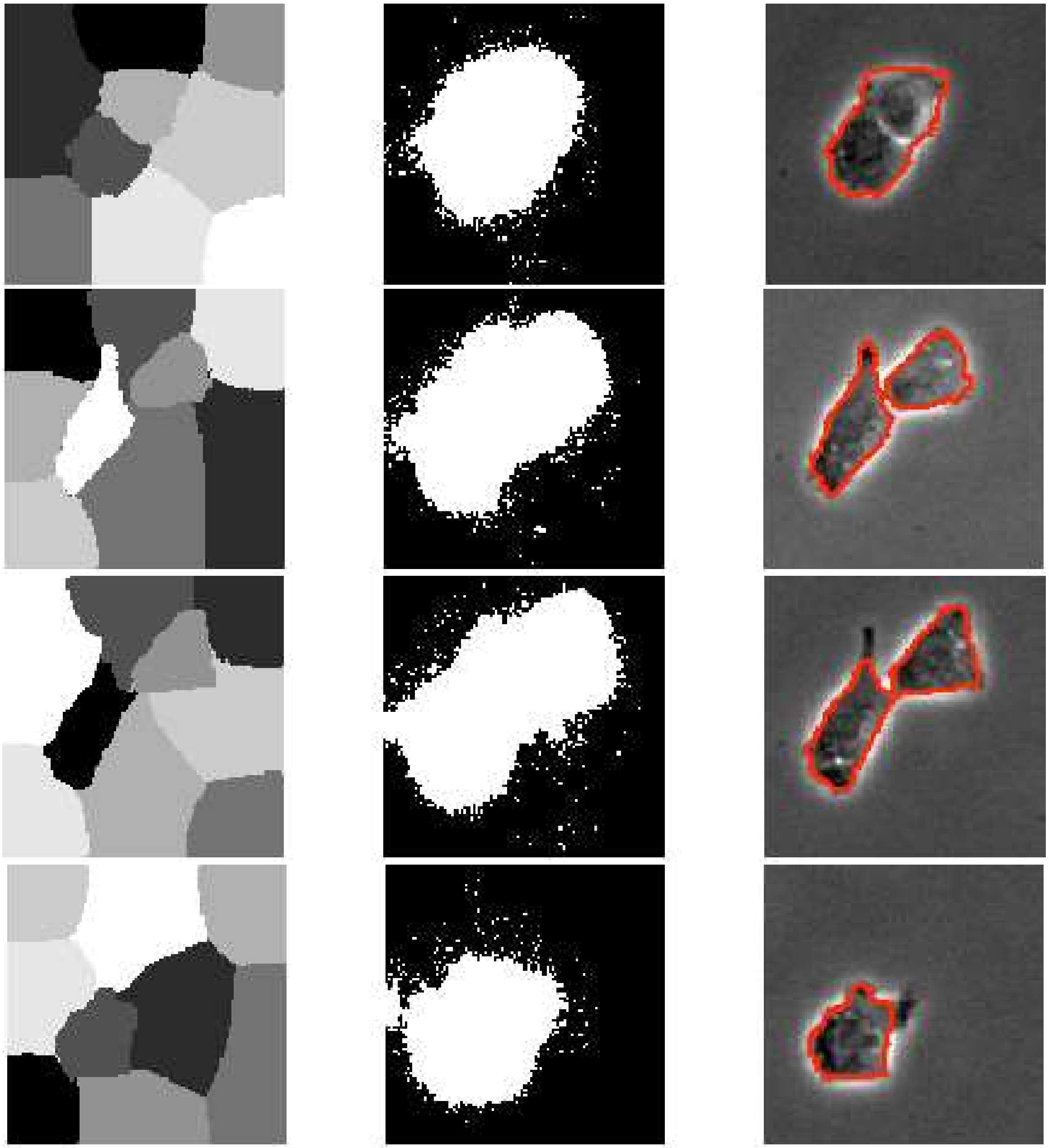

Figure 6.5: Results of the Spectral-K-Means segmentation algorithm

in Figure 6.5, this segmentation algorithm does a good job locating the boundary of cells that are somewhat round in nature, as well as cells that are in the process of dividing (rows 2 and 3). The algorithm was even able to identify two distinct segments for the cell in row 2, which indicates the division has occurred. In Figure 6.6 we can see that the algorithm can do a good job of finding small variations along

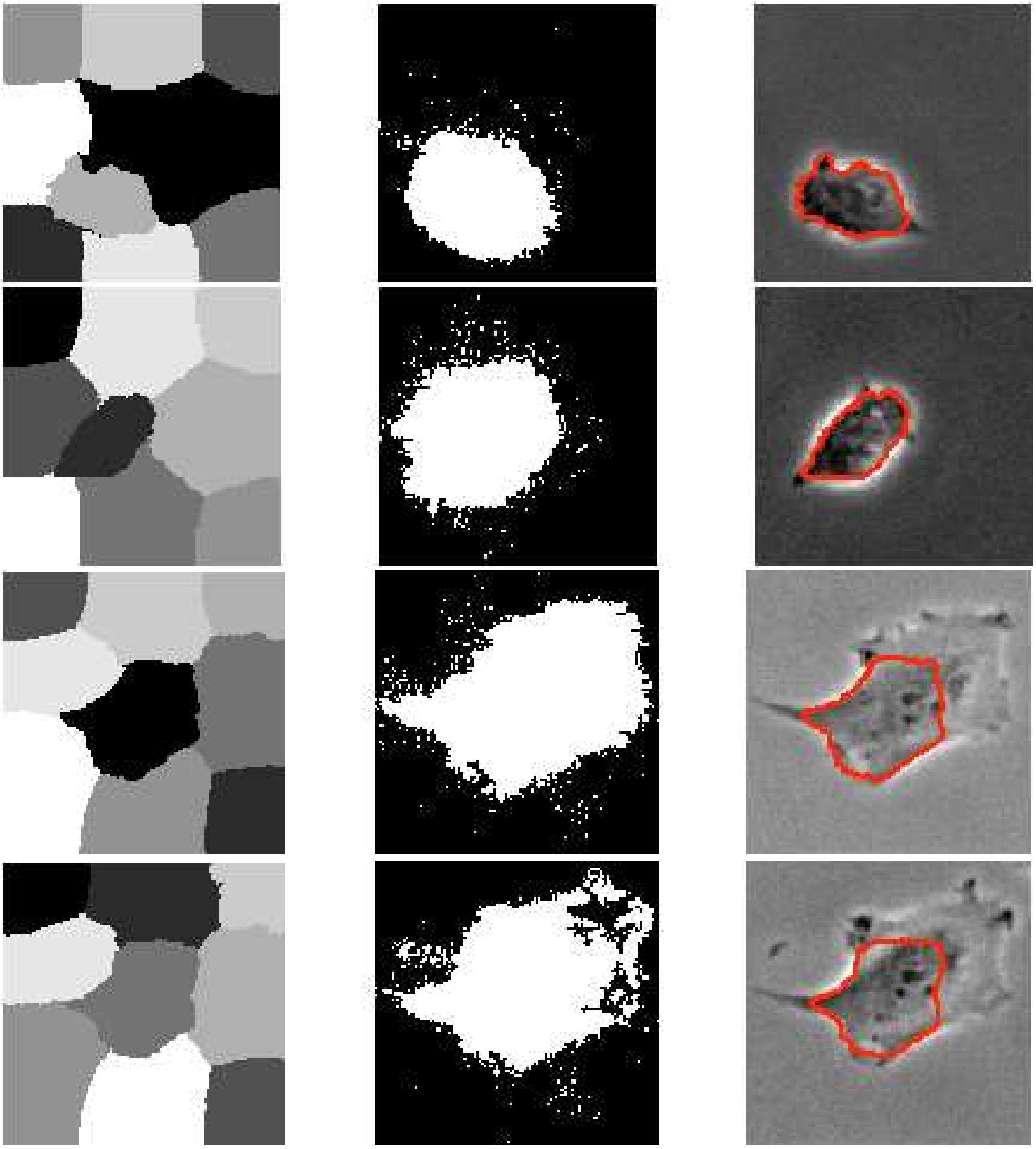

Figure 6.6: Results of the Spectral-K-Means segmentation algorithm

the boundary (row 1), but that it has some trouble with cells that are very close to the background intensity and was only able to find a portion of the cell boundary in rows 3 and 4. Figure 6.7 shows that the algorithm has trouble finding the boundary

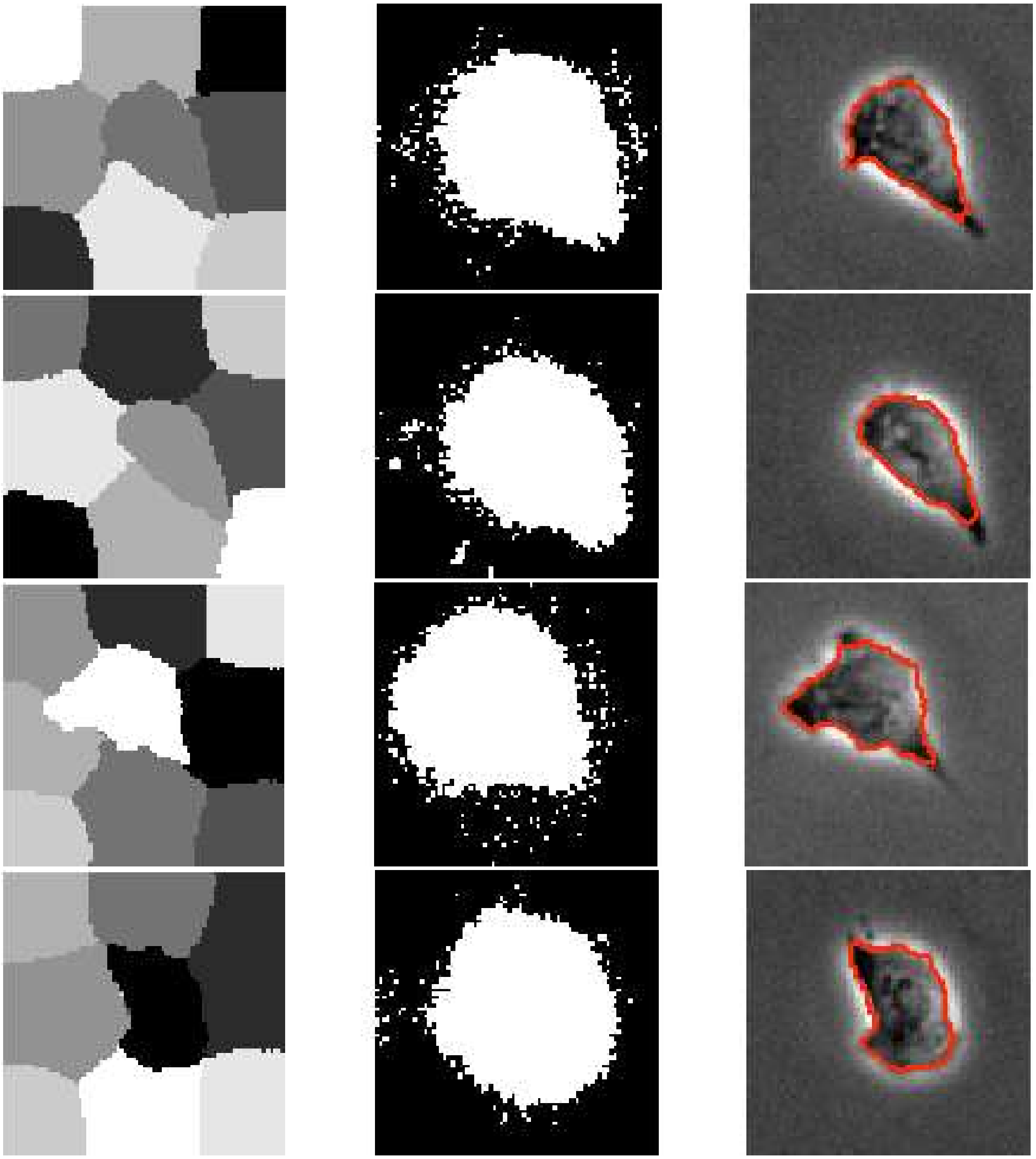

Figure 6.7: Results of the Spectral-K-Means segmentation algorithm

of the pointy tails of the cells. And finally in Figure 6.8 we see that longer cells are somewhat of an issue for the algorithm. This may be attributed to the nature of the distance measure for K-means. The graph generated from the images of the cells weights the edges based partially on a distance measure. This may naturally lend itself to round segments when using the K-Means algorithm. Another thing

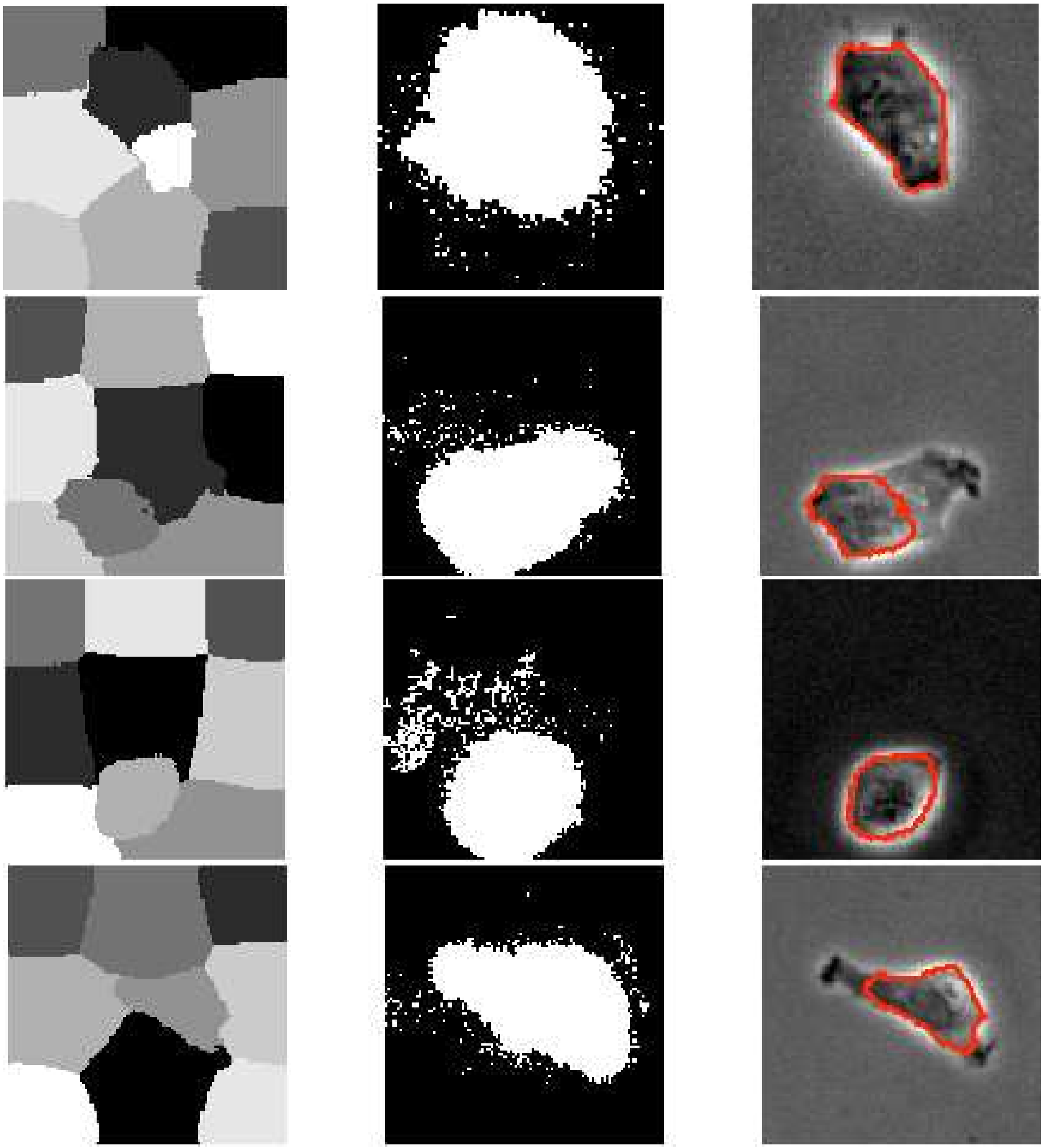

Figure 6.8: Results of the Spectral-K-Means segmentation algorithm

to note is that sometimes a cell is made up of more than one segment from the K-Means partitioning, as in row 1 of Figure 6.8. This is easily taken care of by the second step of the algorithm since both segments fall within the approximate cell area.

6.4 Images Containing Multiple Cells

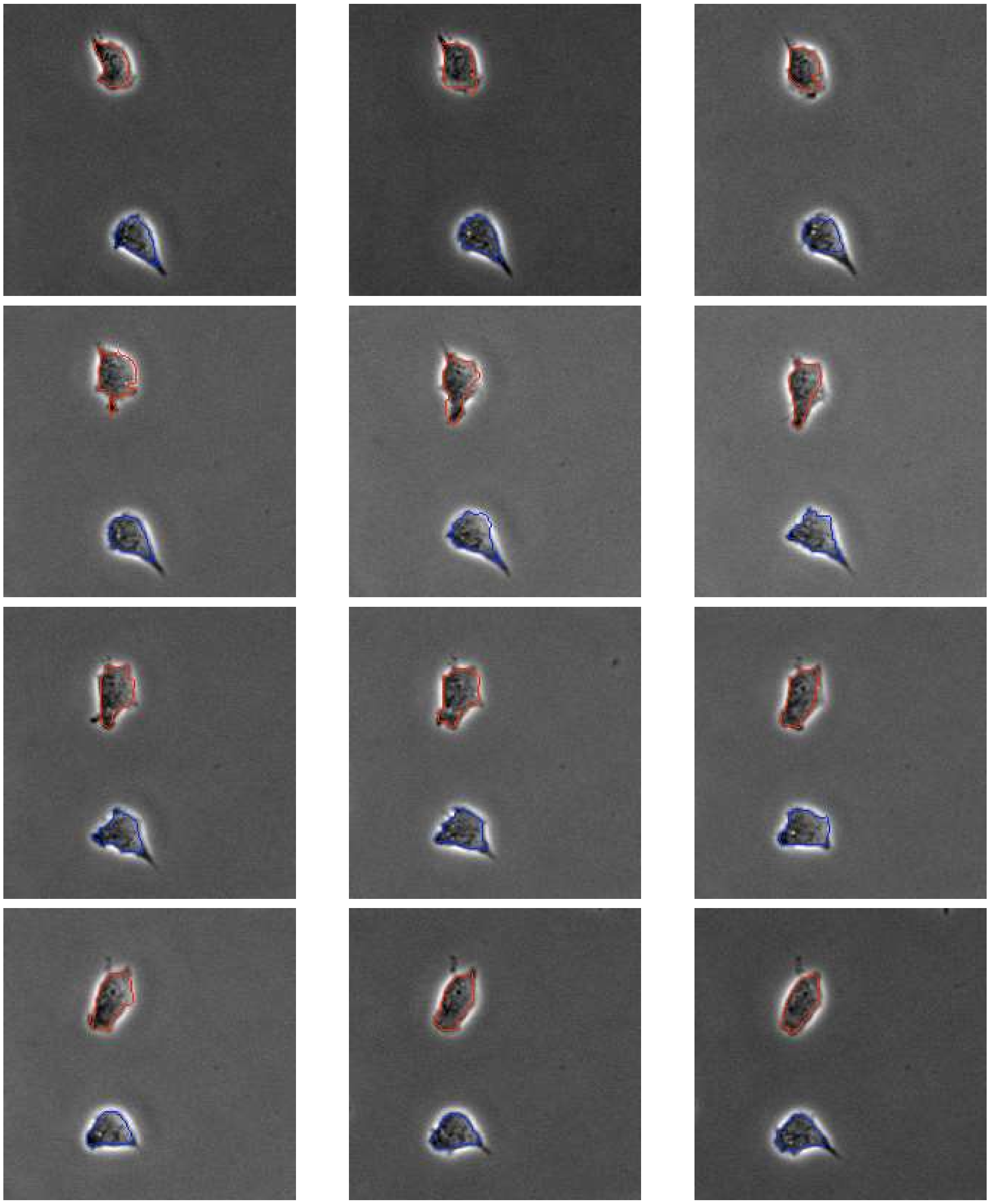

Figure 6.9: Two Cell Automatic Segmentation using Multiphase Watershed Algorithm

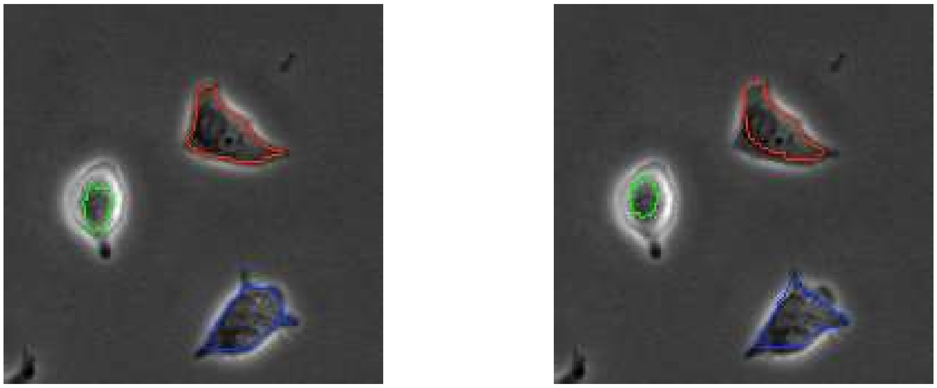

Figure 6.10: Three Cell Automatic Segmentation using Multiphase Watershed Algorithm

Part of this algorithm is to be able to individually segment cells and to be able to follow a cell through a series of images. The algorithm is described in Chapter 5. Figure 6.9 shows a series of images that segment two cells. The windows containing the cells are calculated manually in the first image (top left), but the algorithm calculates them automatically after the first image. These images were generated using the multicell framework and the Multiphase Watershed algorithm for segmentation. The images run from the top left to the bottom right.

We can see from these images that the algorithm is able to handle two cells that are contained within the same frame of an image, and is able to keep track of which cell is which for the duration of the segmentation.

Figure 6.10 shows the results of the same algorithm described above working on an image of three cells. We notice in this image that the multiphase watershed algorithm has some trouble with small bright cells, as seen in the cell segmented with the green boundary.

Finally Figure 6.11 shows an automatic multiple cell segmentation using three cells and the Spectral-K-Means segmentation algorithm. It also shows that the multiple cell segmentation framework is able to work well no matter what segmentation algorithm is used for the segmenting of the cells.

While images containing only two or three cells are shown here, the algorithm naturally extends to the case where there are more than three cells.

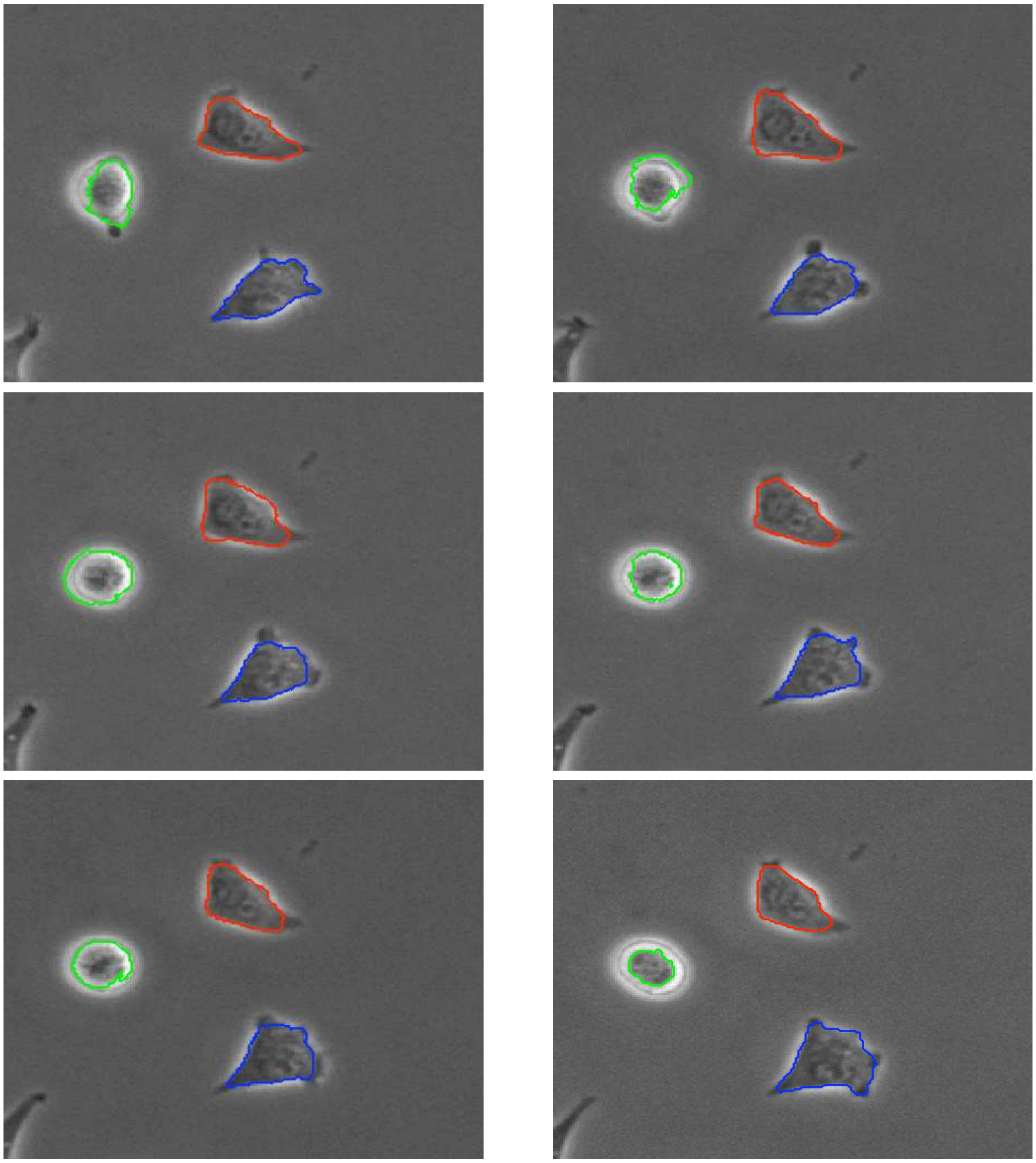

Figure 6.11: Three Cell Automatic Segmentation using Spectral K-means Algorithm

Chapter 7

Conclusions

This paper has demonstrated the challenges of segmenting bright-field images of cells, such as the broken boundary, poor contrast, partial halo and overlapping cells. These challenges are not all present in other types of cell images such as fluorescent images.

There are several papers that solve the segmentation problem for other types of cell images, and some that present a solution for the segmentation of bright-field cell images. The solutions presented for other cell image types do not work well for bright-field images due to the complexities that are specific to the bright-field image. Many of the solutions presented for bright-field images are either only semi-automatic and require human interaction, require the images to be captured in a very specific way or are not able to track cells through frames of images.

I presented three possible solutions. The first was based on a multiphase level set method and the watershed method. The resulting algorithm was able to provide good segmentations of many of the cell images presented, but required some hueristic methods to separate the background from the cell for the seed for the watershed method. The second solution was also based on level set methods, but concentrated on pulling apart statistics of the two regions (background and object). This solution was not able to find the cell boundaries in the images. The final solution was based on spectral graph theory, K-Means and the separation of variances. This method was also able to segment many complex bright-field images.

Both of the working segmentation algorithms presented were also able to segmented images containing multiple cells working with the presented multiple cell segmentation framework.

Possible future work includes further testing of the the robustness of both of the presented methods and extending the multiple cell framework to work with images of the cells that are touching.

References

[1] R. Ali, M. Gooding, M. Christlieb, and M. Brady. Phase-based segmentation of cells from brightfield microscopy. *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 57–60, April 2007. 22

[2] R. Ali, M. Gooding, M. Christlieb, and M. Brady. Advanced phase-based segmentation of multiple cells from brightfield microscopy images. *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 181–184, May 2008. 22

[3] W. Beaver, D. Kosman, G. Tedeschi, E. Bier, W. McGinnis, and Y. Freund. Segmentation of nuclei in confocal image stacks using performance based thresholding. *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 53–56, April 2007. 21

[4] T.F. Chan and L.A. Vese. Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, Feb 2001. ix, 7, 8, 9, 10, 11

[5] Fan R. K. Chung. *Spectral Graph Theory.* American Mathematicsl Society, United States of America, second edition, 1997. 15, 28

[6] O. Debeir, I. Adanja, N. Warzee, P. Van Ham, and C. Decaestecker. Phase contrast image segmentation by weak watershed transform assembly. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 724–727, May 2008. 20

[7] O. Dzyubachyk, W. Niessen, and E. Meijering. Advanced level-set based multiple-cell segmentation and tracking in time-lapse fluorescence microscopy images. *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 185–188, May 2008. 19, 20

[8] Frdric Gibou and Ronald Fedkiw. A fast hybrid k-means level set algorithm for segmentation. Technical report, 4th Annual Hawaii International Conference on Statistics and Mathematics, 2002.

[9] Gerald Karp. *Cell and Molecular Biology: Concepts and Experiments.* John Wiley and Sons, New Jersey, USA, fourth edition, 2005. 2, 20, 21

[10] Anna Korzynska, Wojciech Strojny, Andreas Hoppe, David Wertheim, and Pawel Hoser. Segmentation of microscope images of living cells. *Pattern Anal. Appl.*, 10(4):301–319, 2007. 3, 21

[11] M. Marcuzzo, P. Quelhas, A. Campilho, and A.M. Mendonca. Automatic cell segmentation from confocal microscopy images of the arabidopsis root. In *Biomedical Imaging: From Nano to Macro, 2008. ISBI 2008. 5th IEEE International Symposium on*, pages 712–715, May 2008. 21

[12] Stanley Osher. Level set methods. In *Geometric Level Set Methods in Imaging, Vision, and Graphics*, pages 3–20. Springer New York, 2003. 5

[13] J.A. Sethian. *Level Set Methods and Fast Marching Methods.* Cambridge University Press, United States of America, second edition, 1999. 5

[14] L. Shafarenko, M. Petrou, and J. Kittler. Automatic watershed segmentation of randomly textured color images. *Image Processing, IEEE Transactions on*, 6(11):1530–1544, Nov 1997. 15, 23

[15] Tse Shu Tong. Personal communication, 2007. 23

[16] D.A. Tolliver and G.L. Miller. Graph partitioning by spectral rounding: Applications in image segmentation and clustering. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1053–1060, June 2006. 15, 28

[17] L.A. Vese and T.F. Chan. A multiphase level set framework for image segmentation using the mumford and shah model. *International Journal of Computer Vision*, 50(3):271–293, Dec 2002. ix, 11, 12, 23, 31

[18] Meng Wang, Xiaobo Zhou, Fuhai Li, J. Huckins, R.W. King, and S.T.C. Wong. Novel cell segmentation and online learning algorithms for cell phase identification in automated time-lapse microscopy. In *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pages 65–68, April 2007. 3, 20

[19] Kenong Wu, D. Gauthier, and M.D. Levine. Live cell image segmentation. *Biomedical Engineering, IEEE Transactions on*, 42(1):1–12, Jan. 1995. 22

[20] Anthony Yezzi, Andy Tsai, and Alan Willsky. A statistical approach to curve evolution for image segmentation. Technical report, MIT LIDS Technical Report, Jan 1999. 25

[21] Anthony Yezzi, Andy Tsai, and Alan Willsky. A fully global approach to image segmentation via coupled curve evolution equations. *Journal of Visual Communication and Image Representation*, 13:195–216, 2002. 13, 25, 26