Automated Segmentation of Brightfield Images using the Bhattacharyya Measure via Graph-Cut Segmentation

by

Soo Min Kang

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

Supervisor: Prof. Justin W.L. Wan

Waterloo, Ontario, Canada, 2012

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

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

Abstract

Segmentation of brightfield microscopy images can provide useful information about the behaviour of the cells to biologists. However, its features: poor contrast, broken halo, and missing boundaries, bring great challenges for programmers to develop an automated algorithm that can detect the boundaries of the cells as accurately as the human eye.

In this paper, we present two algorithms based on graph-cut segmentation via Bhattacharyya measure to segment cells in brightfield images. The first model, referred to as the global-local technique, is a two-step procedure. It begins by segmenting the entire image to obtain a global information, the location of isolated cells or groups of cells. It is then followed by a segmentation process that focuses on individual blobs to obtain a refined segmentation result of that specified blob. The second model, referred to as the λ_1-λ_2 technique, is also a two-step process. It uses the effect of under- and over-segmentation that result from two different parameters, λ_1 and λ_2, to obtain a result that is in-between. The effectiveness of both methods are demonstrated on C2C12 cells of brightfield images.

Acknowledgements

I would like to thank Justin W.L. Wan for all his support and guidance with this project. I would also like to thank Robert Sladek and Haig Djambazian from the Department of Medicine and Human Genetics at McGill University for generating the C2C12 microscopy images. Last but not least, I would like to thank my beloved sister, Minsung Kang, my parents, my friends, and Raymond Borja, for all their support and encouragement.

Dedication

This paper is dedicated to my family and the advancements of image processing.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

All living organisms are made of cells. By studying its behaviour, biologists can attain valuable information to aid them in the development and discovery of tissue engineering, stem cell research, drugs, and other related fields [15]. Behaviours of cells, which includes movement, change in area and shape over time, can be studied by observing the position and/or the outline of the cell using a time-lapse microscopy. Time-lapse microscopy is a way of photographing a set of objects at regular intervals for a fixed period of time using a microscope. Each experiment produces hundreds of images, referred to as a frame, with tens to hundreds of cells per frame. Manual detection and segmentation of each cell can be tedious and error prone. Thus, a way to find the outlines of cells automatically, or semi-automatically, has been of great interest to many researchers in the field of medical imaging for several years [14].

Image segmentation is a process of locating the boundaries of objects in a given image. One of the most challenging problems that programmers face is developing a segmentation method that can detect edges of an object as accurately as the human eye [11]. Some common cell images that have been studied in this field are fluorescent and brightfield images (see Figure 1.1). Fluorescent and brightfield images have their own advantages and disadvantages to biologists and image processors. These will be discussed in the rest of this chapter.

Fluorescent images are produced by labelling cells with protein markers that will fluoresce when exposed to light of a particular wavelength. When a cell is successfully fluoresced, a bright, convex shaped object that can easily be distinguished from its background appears [8][13]. This high contrast feature of fluorescent images brought success to various segmentation techniques, such as level set segmentation (see Figure 1.2 left) [8]. However,

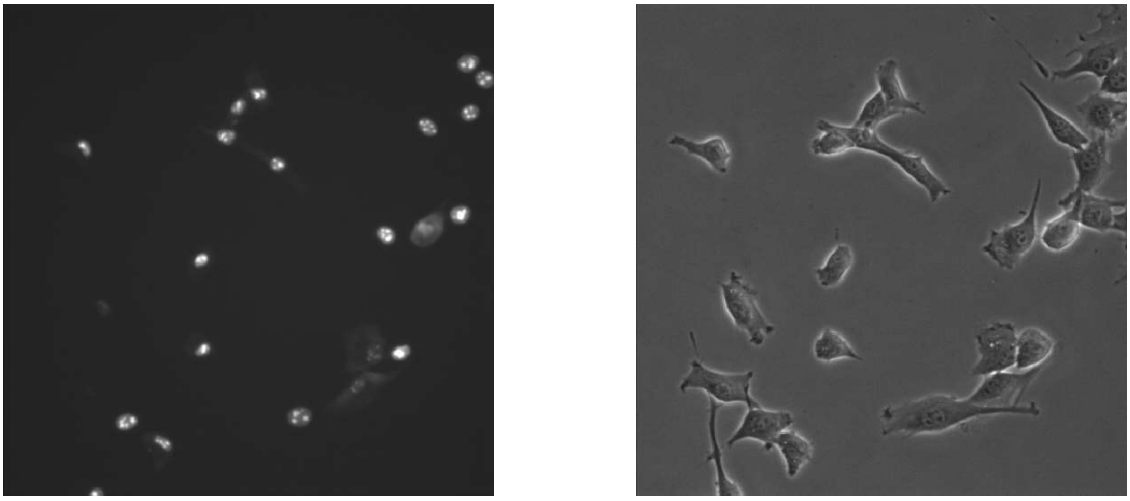


Figure 1.1: A frame of cells imaged using fluorescent microscopy (left) and brightfield microscopy (right).

fluorescent markers are object-specific. That is, only those organelles, such as the nucleus, that are labelled with special proteins will fluoresce under a fluorescent microscope. For experiments that require information on the morphology of the cells, fluorescent images alone are insufficient. In addition, studies have shown that the agents that are used to fluoresce molecules can cause unwanted damage or destruction of the cell [1][12]. Furthermore, disappearance and reappearance of cells have been of great issue in fluorescent images. This forces biologists to guess if a cell that has made a sudden appearance in the middle of an experiment is a result of cell differentiation or limitation of fluorescent microscopy imaging. Although fluorescent images are suitable for some biological studies, its aforementioned features makes them insufficient to study the change in shape of the entire cell, as well as an unreliable way to track the behaviour of a cell, over time [16][17].

Brightfield microscopy is one of the other microscopy techniques that is commonly used today. In brightfield microscopy, a sample is illuminated from below with white light, and observed from above. When light is emitted, the complex structure of a cell absorbs light to various degrees. This results in a non-uniform blotches of dark and bright spots depicting a cell on a uniformly gray background (see Figure 1.1 right). As light tries to pass through the cell membrane, its thickness distorts and reflects the path of the light. Distortion results in a black belt portraying the border of the cell, and the reflection creates a brightened light around the border called a halo, giving emphasis of the cell's boundary on a brightfield image [14]. Unfortunately, a well-connected halo is not present in all cells

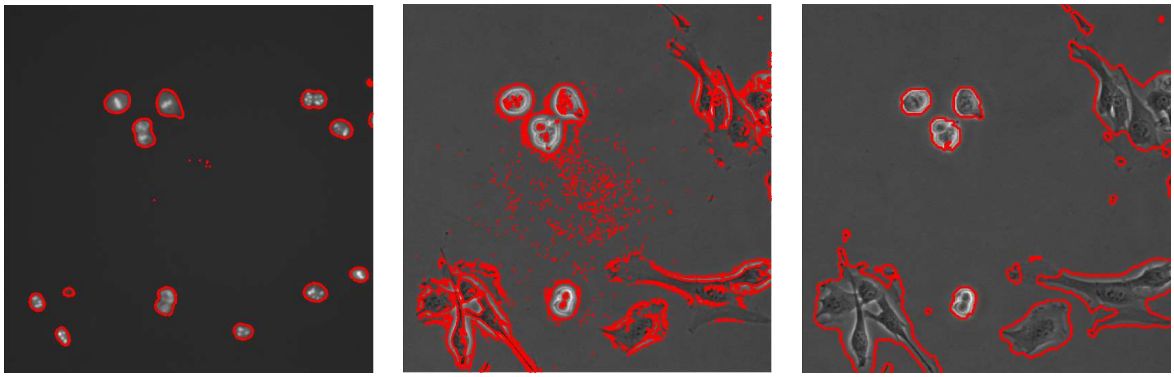


Figure 1.2: Level set and graph-cut method on fluorescent and brightfield images. Left: Level set method performed on a fluorescent image. Centre: Level set method performed on a brightfield image. Right: Graph-cut method performed on a brightfield image.

of brightfield images. Only those that are round have a relatively clear outline with respect to its background. Others, such as flattened cells, tend to have broken boundaries and/or partial halos because the thickness of the membrane is not able to distort and reflect enough light to create a distinguishable cell-to-background contrast. Although brightfield images provide enough information for biologists to study cell's morphology and is a reliable way to track a cell's activity, the low contrast between the background and the cells, as well as its discontinuous to absent halo brings great challenges for programmers to segment (see Figure 1.2 centre).

The difficulties involved with segmenting brightfield images have challenged many researchers in the field of image processing. Numerous work on segmentation of brightfield images have been published with a broad range of approaches. Some models use the solution of the transport of intensity equation to produce high-contrast images. These high-contrast images are then segmented using basic segmentation methods. However, various results found that this technique is sensitive to noise. In [18], Wu et al. presented a statistical approach by using a thresholded variance map. This approach is not applicable to images containing multiple cells, and requires specific modifications of the image data. In [8], Bradbury presented an approach based on K-means clustering and spectral partitioning to separate variances. However, to segment images with multiple cells, this method required prior information on the location of the cell, and multiple cells could not be segmented synchronously.

The goal of this project is to develop an algorithm that can segment cells in brightfield

images as accurately as the human eye with minimal user input and no prior information of the cell. We present two techniques based on graph-cut approach with a statistical measure, the Bhattacharyya measure, to solve this problem. The first technique, the global-local technique, involves segmenting an image with multiple cells to determine the location of each cell or collection of nearby cells. Each blob is isolated and segmented individually to obtain a detailed result. The second method, the λ_1-λ_2 algorithm, involves using two different sets of parameters to obtain a segmentation.

The remainder of this paper is arranged as follows: Chapter 2 contains background information on graph theory and a brief introduction to a statistical measure, the Bhattacharyya measure; Chapter 3 describes the application of graph theory and Bhattacharyya measure in image processing, followed by two new algorithms that can be used to segment cells in brightfield images; Chapter 4 presents and compares the results of two methods proposed in the previous chapter, and Chapter 5 is the conclusion of this paper.

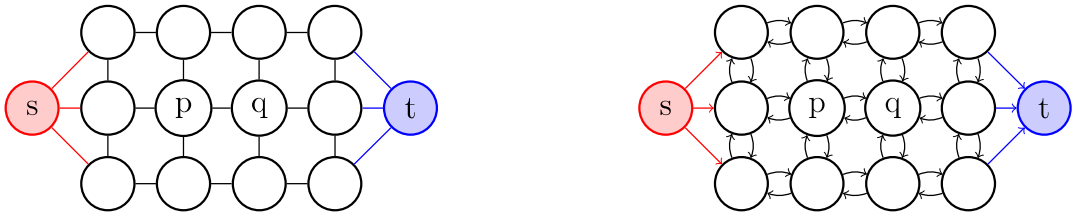
Chapter 2

Background

In this chapter, we describe the background knowledge that is required to understand the formulation that was used in the development of the methods to segment brightfield images. Both methods presented in Chapter 3 uses one of the popular methods in computer vision to segment images, graph-cut. The method presented in this paper differs from other graph-cut approaches, in that it uses a statistical measure known as the Bhattacharyya measure. Therefore, we will give a brief introduction to graph theory, followed by an overview of the Bhattacharyya measure.

2.1 Graph Cut

A graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$ is defined as a set of nodes (vertices) $\mathcal{V}$ and a set of edges $\mathcal{E}$ that connect the nodes. As in Figure 2.1, a graph usually contains two additional special nodes called *terminal nodes*, which are referred to as the source, s, and the sink, t, node. An edge $e \in \mathcal{E}$ that connects an *unordered* pair of nodes $p, q \in \mathcal{V}$ is denoted $e = \{p, q\}$, and is referred to as an *undirected* edge. Edges that connect *ordered* pair of nodes result in two distinct types of edges called *directed* edges, and are denoted (p, q) and (q, p). That is, while an undirected edge $\{p, q\}$ is the same as $\{q, p\}$, directed edges (p, q) and (q, p) are distinct (see Figure 2.1). In this paper, we will make references to undirected edges only, thus all edges will simply be referred to as 'edges' rather than 'undirected edges.' In addition, edges can be further categorized into two types: n-links and t-links. Edges that connect a pair of neighbouring nodes are called *n-links*, while edges that connect nodes to terminal nodes are called *t-links*, where n stands for "neighbours" and t stands for "terminals." All edges in a graph are assigned some weight, or cost, w_e.



(a) Undirected Graph (b) Directed Graph

Figure 2.1: Graphs. Circles represent nodes and lines/arrows represent edges. Coloured nodes, s and t, are terminal nodes, and lines with arrows (right) are directed edges. Coloured edges are t-links and black edges are n-links.

2.1.1 Min-Cut and Max-Flow Problems

An *s-t cut*, or *cut*, partitions the nodes of a graph into two disjoint subsets, $\mathcal{S}$ and $\mathcal{T}$. A cut $\mathcal{C}$ is a set of edges in $\mathcal{E}$, that are involved with separating the nodes into two disjoint subsets, $\mathcal{S}$ and $\mathcal{T}$. That is, an edge $\{p, q\} \in \mathcal{C}$ if $p \in \mathcal{S}$, $q \in \mathcal{T}$, and $\mathcal{S} \cap \mathcal{T} = \emptyset$. In combinatorial optimization, the *cost* (*capacity*) of a cut $\mathcal{C}$, is defined as the sum of the costs of the edges that it severs,

$$|\mathcal{C}| = \sum_{e \in \mathcal{C}} w_e.$$

To solve a *min-cut* problem of a graph, means to find a set of edges that give the smallest cost value among all possible cuts. The solution generates a segmentation that is optimal in terms of the properties that are built into the edge weights, which differ from problem to problem. Chapter 3 will describe a way to assign appropriate weights to segment brightfield images.

Given a sequence of distinct edges that connect s to t, the largest weight among the edges in the set is referred to as *flow*. A determination of *max-flow* to the corresponding graph allows one to find the solution to the min-cut problem with ease using the **max-flow/min-cut theorem**, which states that the value of the maximum flow is equal to the minimum cut capacity [4]. In other words, by finding a flow that saturates a set of edges from s to t, one can find a cut that divides the nodes into two disjoint sets $\mathcal{S}$ and $\mathcal{T}$, which correspond to a minimum cut. Thus, a problem of finding min-cut or max-flow of a graph is referred to as *max-flow/min-cut*.

2.1.2 Algorithms to Compute Max-flow/Min-Cut

There are many algorithms that can compute the maximum flow of a graph, such as augmenting paths [9] and push-relabel [10]. However, implementation of these algorithms tend to be inefficient when applied to vision problems. In [7], Boykov and Kolmogorov introduced an innovative algorithm that was able to solve the max-flow/min-cut problem efficiently. Implementations presented in this paper used the graph-cut algorithm introduced by Boykov and Kolmogorov in [7].

2.2 The Bhattacharyya Measure

To address the issue of low contrast in brightfield images, intensity distributions are carefully considered. The difference in intensity profiles can be detected using the *Bhattacharyya* measure $\mathcal{B}(f, g)$, which evaluates the amount of overlap (or similarity) between two distributions f and g:

$$\mathcal{B}(f, g) = \sum_{z \in \mathcal{Z}} \sqrt{f(z)g(z)}, \tag{2.1}$$

where z is a dependent variable of f and g, and $\mathcal{Z}$ is a set that contains z.

Consider two identical distributions f and g as in Figure 2.2 top-left. If f and g are as in Table 2.1, then the product of the two distributions sum to 1. That is, the Bhattacharyya coefficient of f and g is 1, while the sum of the ratios is equal to the cardinality of $\mathcal{Z}$. Conversely, two non-overlapping distributions f and g as in Figure 2.2 top-right (example depicted in Table 2.2 for some $\varepsilon_1, \varepsilon_2 > 0$ and $\varepsilon_1 < \varepsilon_2$), the Bhattacharyya coefficient tends to 0, and the sum of the ratios approach to infinity as $\varepsilon_1, \varepsilon_2 \to 0$. Thus, the difference between the Bhattacharyya measure and the sum of the ratio is smaller for two similar distributions than of two very different distributions. Although it may be enough to use the Bhattacharyya coefficient alone to determine the similarity of two distributions, since 0 corresponds to no overlap between distributions, and 1 to a perfect match, it can be further compared by observing the discrepancy between the Bhattacharyya coefficient and the sum of the ratios.

z	$f(z)$	$g(z)$	$\sqrt{f(z)g(z)}$	$\sqrt{\frac{f(z)}{g(z)}}$
0	0.005	0.005	0.005	1
1	0.02	0.02	0.02	1
2	0.025	0.025	0.025	1
3	0.05	0.05	0.05	1
4	0.2	0.2	0.2	1
5	0.4	0.4	0.4	1
6	0.2	0.2	0.2	1
7	0.05	0.05	0.05	1
8	0.025	0.025	0.025	1
9	0.02	0.02	0.02	1
10	0.005	0.005	0.005	1
Sum	1	1	1	11

Table 2.1: Example of identical functions f and g.

z	$f(z)$	$g(z)$	$\sqrt{f(z)g(z)}$	$\sqrt{\frac{f(z)}{g(z)}}$
0	0.01	ε_2	$\sqrt{0.01\varepsilon_2}$	$\sqrt{\frac{0.01}{\varepsilon_2}}$
1	0.04	ε_2	$\sqrt{0.04\varepsilon_2}$	$\sqrt{\frac{0.04}{\varepsilon_2}}$
2	0.25	ε_2	$\sqrt{0.25\varepsilon_2}$	$\sqrt{\frac{0.25}{\varepsilon_2}}$
3	0.4	ε_2	$\sqrt{0.4\varepsilon_2}$	$\sqrt{\frac{0.4}{\varepsilon_2}}$
4	0.25	ε_2	$\sqrt{0.25\varepsilon_2}$	$\sqrt{\frac{0.25}{\varepsilon_2}}$
5	0.05	ε_2	$\sqrt{0.05\varepsilon_2}$	$\sqrt{\frac{0.05}{\varepsilon_2}}$
6	ε_1	ε_2	$\sqrt{\varepsilon_1\varepsilon_2}$	$\sqrt{\frac{\varepsilon_1}{\varepsilon_2}}$
7	ε_1	0.15	$\sqrt{0.15\varepsilon_1}$	$\sqrt{\frac{\varepsilon_1}{0.15}}$
8	ε_1	0.7	$\sqrt{0.7\varepsilon_1}$	$\sqrt{\frac{\varepsilon_1}{0.7}}$
9	ε_1	0.15	$\sqrt{0.15\varepsilon_1}$	$\sqrt{\frac{\varepsilon_1}{0.15}}$
10	ε_1	ε_2	$\sqrt{\varepsilon_1\varepsilon_2}$	$\sqrt{\frac{\varepsilon_1}{\varepsilon_2}}$
Sum as $\varepsilon_1, \varepsilon_2 \to 0$	1	1	0	∞

Table 2.2: Example of non-overlapping functions f and g.

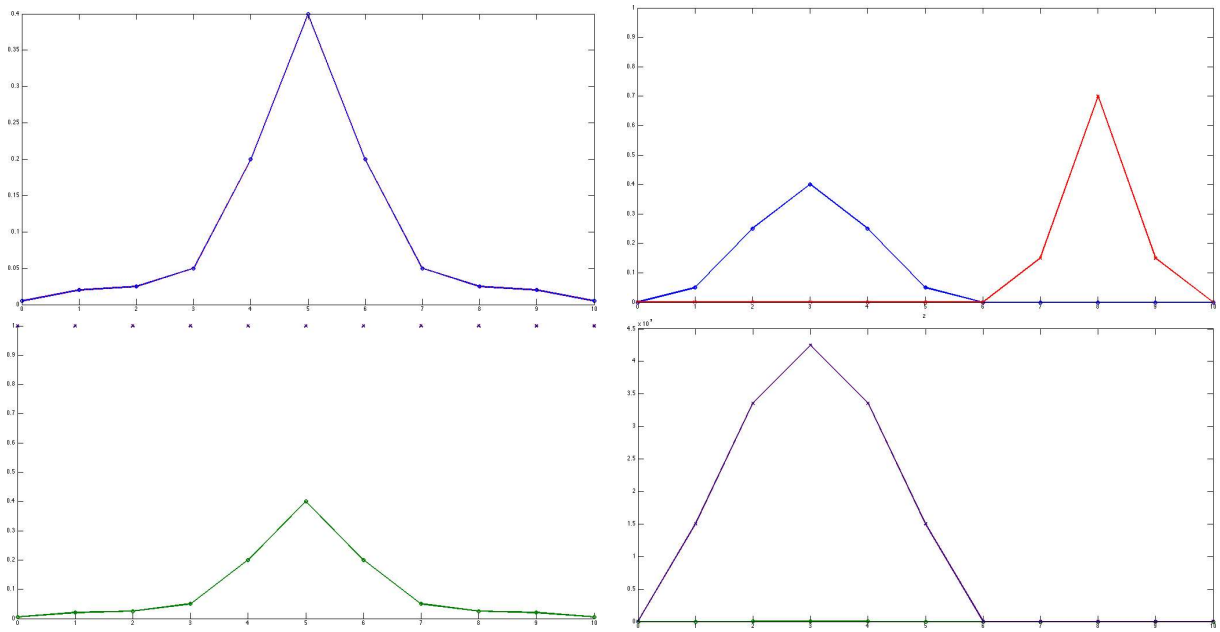


Figure 2.2: Bhattacharyya Measure and Ratio Comparison. The functions f (blue) and g (red) are shown on the top row. $\sqrt{f \cdot g}$ (green) and $\sqrt{\frac{f}{g}}$ (purple) of its respective distributions are displayed in the bottom row. The difference between $\sqrt{f \cdot g}$ and $\sqrt{\frac{f}{g}}$ is less than 1 for identical distributions f and g (bottom-left), while the difference is greater than 10^6 for non-overlapping distributions (bottom-right).

Chapter 3

Methodology

The main algorithm presented in this paper that is used to segment objects from the background of image I, evolves from the Bhattacharyya measure that is introduced in Section 2.2. Suppose p represents the pixel-location of an image I, and I_p contains the intensity value of the image at p. Let L be a binary mask whose elements $L_p \in \{0, 1\}$ describe the labels of p[1]. If we let M to be a learned distribution of the background (or object), and $P_{R_1^L}$ (or $P_{R_0^L}$) a distribution of the regions in I that correspond to $L_p = 1$ (or $L_p = 0$), where $\mathcal{R}_1^L \equiv \{p \in \mathcal{P} | L_p = 1\}$ and $\mathcal{R}_0^L \equiv \{p \in \mathcal{P} | L_p = 0\}$, then we want to find L such that

$$\max_L \mathcal{B}(P_{R_1^L}, M),$$

since larger Bhattacharyya coefficients correspond to more similar distributions, which is equivalent to

$$\min_L -\mathcal{B}(P_{R_1^L}, M). \tag{3.1}$$

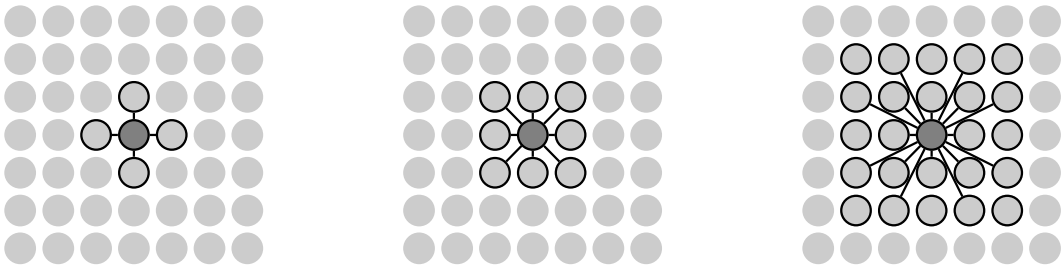
However, optimization of (3.1) has been found to be difficult, and computationally expensive [3]. Thus, Ayed et al. proposed an approach to solve (3.1) by computing a sequence of labels $\{L^{n+1}\}$ using graph-cut optimization.

In the following sections, we will describe the application of graph-cut optimization in image segmentation, followed by the use of the Bhattacharyya measure in graph-cut optimization. This chapter will conclude by presenting two algorithms using graph-cut optimization via Bhattacharyya model to segment cells in brightfield images.

[1] In this paper, $L_p = 0$ implies that p is labelled as the object, and $L_p = 1$ as the background.

3.1 Graph Theory in Image Processing

In the context of using graphs to process images, nodes are often used to depict pixels p of an image I. The two terminal nodes, s and t, correspond to labels "object" and "background," respectively. Edges either provide information about the relationship between its neighbouring pixels, which are typically set up in a grid-like fashion as in Figure 3.1, or give information about its labels. Like the bottom-left diagram of Figure 3.2, a graph is initially set by joining every node to both its terminal nodes, and its neighbouring nodes. A cut severs one of two t-links of p and some n-links to partition the nodes into two disjoint sets $\mathcal{S}$ and $\mathcal{T}$ (see Figure 3.2 bottom-right). The nodes that belong to set $\mathcal{S}$ correspond to pixels that are labelled as "object," and the nodes that belong to set $\mathcal{T}$ represent pixels labelled as "background" (see Figure 3.2 bottom-right to top-right). n-links that are severed as a result of a cut tend to be affected by the boundaries that are near the segmentation of an image. Thus, the weights that are assigned to n-links represent the boundary properties of the segmentation. Severed t-links, on the other hand, represent the labelling, or regional properties, of its segments. Therefore, the minimum cost of a cut corresponds to a segmentation of a balance between boundary and regional properties of an image [5].



(a) 4 n-system (b) 8 n-system (c) 16 n-system

Figure 3.1: Neighbourhood systems. Circles with dark edges represent neighbouring nodes of the dark circle in its corresponding neighbourhood system. The lines between the nodes represent all possible n-links in its neighbourhood system.

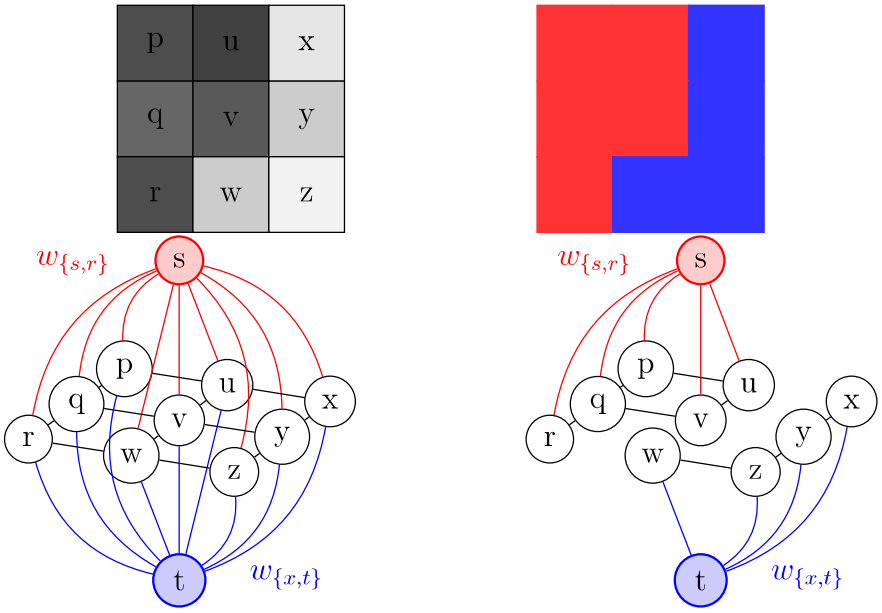


Figure 3.2: Image segmentation via graph-cuts. Top-left: Image given by the user. Bottom-left: Image represented in a form of a graph. Bottom-right: Graph is cut by $\mathcal{C}$, where $\mathcal{C} = \{\{r, w\}, \{v, w\}, \{v, y\}, \{u, x\}\}$. Top-right: Mask of the image that contains labels, object and background, of the regions of the image. (Red represents the "object" label, and blue the "background" label.)

Suppose $\mathcal{P}$ is a set of all pixels in I, and $\mathcal{N}$ a set of all pairs of neighbouring elements under a standard 4-, 8-, 16-, or 24-neighbourhood system (as in Figure 3.1[2]) in $\mathcal{P}$. An energy function to segment an image using boundary and regional properties of L can be defined as:

$$E(L) = \lambda \cdot B(L) + R(L), \tag{3.2}$$

where

$$B(L) = \sum_{\{p,q\} \in \mathcal{N}} B_{\{p,q\}} \tag{3.3}$$

represents the boundary term, which assigns weights to n-links between p and q, and

$$R(L) = \sum_{p \in \mathcal{P}} R_p(L_p) \tag{3.4}$$

[2]The 24-neighbourhood system is not shown in Figure 3.1 due to its complex structure.

represents the regional term, assigning weights to t-links of p. The presence of $\lambda \geq 0$ in (3.2) can specify the relative importance of the regional property $R(L)$ from the boundary property $B(L)$, and vice versa.

3.1.1 Boundaries of L

The weights $B_{\{p,q\}}$ that are assigned on the n-links of p and q act as penalties for the discontinuity between p and q. A separation of neighbouring pixels p and q that belong to the same region, either $p, q \in \mathcal{S}$ or $p, q \in \mathcal{T}$, as a result of a min-cut can be prevented by assigning a large $B_{\{p,q\}}$ value. Conversely, a separation between two neighbouring pixels p and q that belong to different regions, $p \in \mathcal{S}$ and $q \in \mathcal{T}$ or vice versa, can be emphasized for a cut by assigning a small $B_{\{p,q\}}$ value. One of the ways of expressing the cost of $B_{\{p,q\}}$ is:

$$B_{\{p,q\}} \propto \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right). \tag{3.5}$$

In (3.5), if I_p and I_q contain the same intensity values, that is, $I_p - I_q = 0$, then $B_{\{p,q\}}$ is maximized for some constant σ. On the contrary, a pair of very different pixel intensities is assigned a small weight to edge $\{p, q\}$. That is, $B_{\{p,q\}} \to 0$ as $I_p - I_q \to \infty$.

Furthermore, the penalty of p and q can be strengthened for neighbouring nodes that are close together by including a distance function of p and q in the definition of $B_{\{p,q\}}$ as in [5]:

$$B_{\{p,q\}} \propto \frac{1}{\|p - q\|} \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right). \tag{3.6}$$

3.1.2 Regions of L

Since every node in a graph is initially connected to both terminals, s and t, prior to a cut, weights that connect to the two terminals must be considered simultaneously. Say $w_{\{s,p\}}$ and $w_{\{p,t\}}$ denote the weights of edges that connect node p to s, and p to t, respectively. Then $R(L)$ in (3.4) can be written as:

$$R(L) = \sum_{p \in \mathcal{P}} w_{\{s,p\}} + \sum_{p \in \mathcal{P}} w_{\{p,t\}}. \tag{3.7}$$

A cut removes exactly one of the two t-links, preferably the link with a smaller weight. As in Figure 3.3, if $w_{\{s,p\}} > w_{\{p,t\}}$, then min-cut severs the t-link that connects p to t, and p

belongs to set $\mathcal{S}$, which is a set that contains the terminal node s but not t. If a t-link that connects node q to s, on the other hand, weighs less than the t-link that connects q to t ($w_{\{s,q\}} < w_{\{q,t\}}$), then $\{s, q\}$ is disconnected, and $q \in \mathcal{T}$. Mathematically speaking, prior to min-cut, $\mathcal{S} = \{s\}$, $\mathcal{T} = \{t\}$, and $\mathcal{P} = \{\dots, p, q, \dots\}$. If $w_{\{s,p\}} > w_{\{p,t\}}$ and $w_{\{s,q\}} < w_{\{q,t\}}$, then $\mathcal{S} = \{s, p, \dots\}$, $\mathcal{T} = \{t, q, \dots\}$, and $\mathcal{P} = \emptyset$ after min-cut occurs, where $\mathcal{S} \cap \mathcal{T} = \emptyset$ before and after min-cut. Therefore, like n-links, assigning large weights to edges that are likely to belong to one set than the other, will lessen the likelihood of that t-link being cut. The weights that are assigned on t-links of p act as penalties for labelling p as "object" or "background."

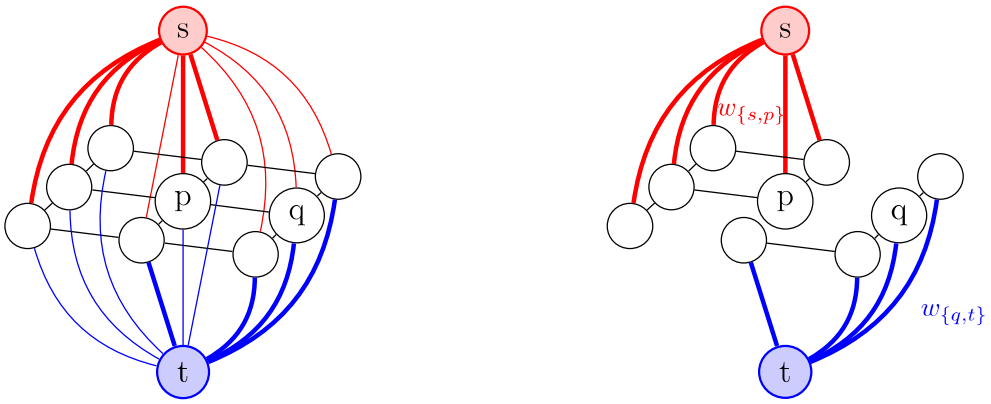


Figure 3.3: Min-cut. Thickness of t-links indicate the relative weights of each edge. That is, $w_{\{s,p\}} > w_{\{p,t\}}$ and $w_{\{q,s\}} < w_{\{q,t\}}$. Left: A weighted graph before it is cut. Right: A cut separates pixels into two distinct sets $\mathcal{S}$ and $\mathcal{T}$ such that $p \in \mathcal{S}$ and $q \in \mathcal{T}$, where $s \in \mathcal{S}$, $t \in \mathcal{T}$, and $\mathcal{S} \cap \mathcal{T} = \emptyset$.

Many methods that segment images via graph-cuts use learned histograms to define weights for t-links. In this project, we follow the approach by Ayed et al., and use the Bhattacharyya measure to assign weights for t-links. Note that due to the nature of brightfield images, which has extremely low contrast between cells and the background, we carefully chose to use the intensities of the background when constructing the intensity distribution rather than the cells since cells tend to have more variations in intensity (see Figure 3.4). Thus, the learned distribution M is of the background. Correspondingly, the distribution of the regions in I to be compared to in the Bhattacharyya measure are those regions that correspond to $L_p = 1$, the background. Before we proceed on to define the penalties for labelling nodes as "object" or "background," we introduce some definitions and notations.

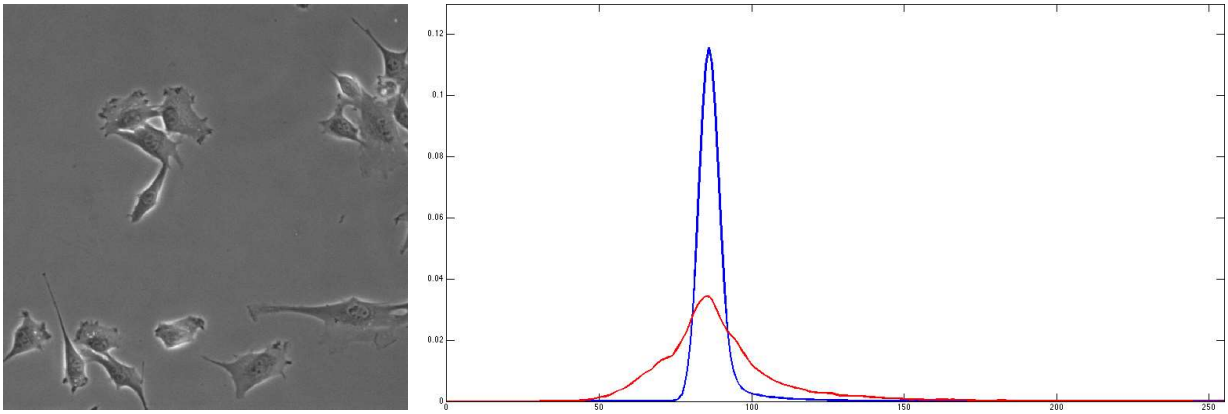


Figure 3.4: Distribution Comparison: Cells vs. Background. Intensity distribution of image on left is shown on right with cells in red and background in blue. The width of the distributions indicate the variance of intensity per region. The narrow width of the blue distribution indicates that the background has less variable intensity.

Let $\mathcal{Z}$ be a set of all numerical values that I_p can take. For example, if I_p represents an intensity value that ranges from 0 to 255, then $\mathcal{Z} = \{z \in \mathbb{N} \mid 0 \leq z \leq 255\}$. Let $A(\mathcal{R})$ denote the number of pixels in region $\mathcal{R}$, that is, $A(\mathcal{R}) = \sum_{\mathcal{R}} 1$, and $K_z(I_p)$ the Gaussian kernel of I, which is mathematically written as:

$$K_z(I_p) = \frac{1}{\sqrt{2\pi\omega^2}} \exp\left(-\frac{(z - I_p)^2}{2\omega^2}\right) \forall\ z \in \mathcal{Z} \tag{3.8}$$

where $\omega > 0$ is the width of the kernel. The value of ω in (3.8) determines the smoothness of the distribution. As $\omega \to 0$, the Gaussian kernel (3.8) approaches the Dirac function:

$$K_z(I_p) = \begin{cases} \infty & \text{if } z = I_p \\ 0 & \text{if } z \neq I_p \end{cases},$$

a discrete normalized histogram, while $\omega > 0$ corresponds to a smooth, continuous normalized histogram. Note that large ω values complement smooth Gaussian distributions.

$P_{R_1^L}$ is formally defined as a kernel density estimate (KDE), which is very closely related to histograms. KDE is a sum of kernels, such as Gaussians, producing a smoothened histogram. A KDE $P_{R_1^L}(z)$ of image I in region $\mathcal{R}_1^L$ can be defined as:

$$P_{R_1^L}(z) = \frac{\sum_{p \in \mathcal{R}_1^L} K_z(I_p)}{A(\mathcal{R}_1^L)} \forall\ z \in \mathcal{Z}.$$

Following the approach by Ayed et al., we assign the terminal weights the following way:

$$w_{\{s,p\}} = (1-\alpha)\left(\frac{-\mathcal{B}(P_L, \boldsymbol{M})}{A(\mathcal{R}_1^L)}\right), \text{ and} \tag{3.9}$$

$$w_{\{p,t\}} = L_p\left(\frac{-\mathcal{B}(P_L, M)}{A(R_1^L)} + \sum_{z\in Z}\frac{K_z(I_p)}{A(R_1^L)}\sqrt{\frac{\boldsymbol{M}(z)}{P_L(z)}}\right), \tag{3.10}$$

where $\mathcal{B}(\cdot,\cdot)$ is the Bhattacharyya measure as described in Section 2.2 for $\alpha \in [0,1]$. Thus, the weights for the t-links defined in (3.9) and (3.10) can be substituted into (3.11) such that

$$R(L) = (1-\alpha)\sum_{p\in R_0^L}\frac{-\mathcal{B}(P_L, \boldsymbol{M})}{A(\mathcal{R}_1^L)} + \sum_{p\in R_1^L} L_p\left(\frac{-\mathcal{B}(P_L, M)}{A(R_1^L)} + \sum_{z\in Z}\frac{K_z(I_p)}{A(R_1^L)}\sqrt{\frac{\boldsymbol{M}(z)}{P_L(z)}}\right). \tag{3.11}$$

Relationship between the t-links

One may notice that the two t-links defined in (3.9) and (3.10) are dependent solely on the comparison of background distributions, rather than of both background and object distributions. Although this may seem strange at first, the relationship between the two terminal weights as defined in (3.9) and (3.10) can be seen by observing the difference between the two for all those pixels that are labelled as "background." First, we note that

$$\begin{aligned} & w_{\{s,p\}} - w_{\{p,t\}} \\ =& (1-\alpha)\left(\frac{-\mathcal{B}(P_L, M)}{A(R_1^L)}\right) - \left(\frac{-\mathcal{B}(P_L, M)}{A(R_1^L)} + \frac{1}{A(R_1^L)}\sqrt{\frac{M(I_p)}{P_L(I_p)}}\right) \quad \forall p \in \mathcal{R}_1^L \\ =& \frac{1}{A(R_1^L)}\left(\alpha\mathcal{B}(P_L, M) - \sqrt{\frac{M(I_p)}{P_L(I_p)}}\right). \end{aligned} \tag{3.12}$$

Two cases follow:

1. $0 < w_{\{s,p\}} - w_{\{p,t\}}$
2. $0 > w_{\{s,p\}} - w_{\{p,t\}}$

If $0 < w_{\{s,p\}} - w_{\{p,t\}}$, then for all $p \in R_1^L$, p must be removed from the current "background" region, which can be done by setting L_p to 0. This forces the $\sqrt{M(I_p)/P_L(I_p)}$ value of

(3.12) to increase, reducing the difference between $\sqrt{M/P_L}$ and $\alpha\mathcal{B}(P_L, M)$ corresponding to more alike distributions (see Section 2.2). Conversely, if $0 > w_{\{s,p\}} - w_{\{p,t\}}$, then p must remain in the "background" region. Otherwise, the discrepancy between $\mathcal{B}(P_L, M)$ and $\sqrt{M(I_p)/P_L(I_p)}$ would be increased, which is unfavourable (see Section 2.2).

Hard Constraints

In many real images, especially brightfield images, objects do not have sufficiently distinct regional properties. This causes the algorithm to label nodes incorrectly especially those near the boundary of the segmentation. If pixel p is known to belong to one class, say "object," then by assigning the t-link that connects the p node to the sink node s (node that corresponds to the "object" label) with a cost of infinity would enforce the algorithm to label p as "object" rather than "background" when it terminates after finding min-cut of the graph [5]. Boykov and Jolly [6] referred to this constraint as a "hard constraint," and emphasized the importance of including this constraint within the energy function when computing the global minimum of (3.2). In this project, we restrict the use of hard constraints on "objects" only, to emphasize the growth of the "object" region as the algorithm proceeds. This can be done by adding some large value K to the weight of the t-link that joins p to the terminal node s, which labels p as "object":

$$w_{\{s,p\}} = (1-\alpha)\left(\frac{-\mathcal{B}(P_L, M)}{A(\mathcal{R}_1^L)}\right) + K(1 - L_p).$$

Finding the solution to the Energy function

Solving the energy functional (3.2) with $B(L)$ and $R(L)$ defined as (3.6) and (3.11) can be computationally expensive. An alternative way to find L is by computing a sequence of labels $\{L^{n+1}\}$, which are solutions of the minimization problems parametrized by α_n:

$$\begin{aligned} L^{n+1} = \arg\min_L \Bigg\{ & \lambda B(L) + (1-\alpha_n) \sum_{p \in R_0^L} \frac{-\mathcal{B}(P_{L^n}, \boldsymbol{M})}{A(\mathcal{R}_1^{L^n})} \\ & + \sum_{p \in R_1^L} L_p^n \left(\frac{-\mathcal{B}(P_{L^n}, M)}{A(R_1^{L^n})} + \sum_{z \in Z} \frac{K_z(I_p)}{A(R_1^{L^n})} \sqrt{\frac{\boldsymbol{M}(z)}{P_{L^n}(z)}} \right) + K(1 - L_p^n) \Bigg\}. \end{aligned} \tag{3.13}$$

As $\alpha_n \to 0$, L^n converges to the solution of $\min_L \lambda B(L) - \mathcal{B}(P_L, \boldsymbol{M})$ [2].

3.1.3 Summary

In summary, by solving the optimization problem:

$$\min_{L} \lambda B(L) + R(L)$$

where $B(L) = \sum_{\{p,q\} \in \mathcal{N}} w_{\{p,q\}}$ and $R(L) = \sum_{p \in \mathcal{P}} w_{\{s,p\}} + \sum_{p \in \mathcal{P}} w_{\{p,t\}}$ whose weights are summarized in Table 3.1, one can generate a binary labelling mask L of an image.

Edge e	Weight w_e
$\{p,q\}$	$\frac{1}{\|p-q\|} \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right)$
$\{s,p\}$	$(1-\alpha_n)\left(\frac{-\mathcal{B}(P_{L^n}, \boldsymbol{M})}{A(\mathcal{R}_1^{L^n})}\right) + K(1 - L_p^n)$
$\{p,t\}$	$L_p^n\left(\frac{-\mathcal{B}(P_{L^n}, M)}{A(R_1^{L^n})} + \sum_{z \in Z} \frac{K_z(I_p)}{A(R_1^{L^n})} \sqrt{\frac{\boldsymbol{M}(z)}{P_{L^n}(z)}}\right)$

Table 3.1: Weights.

3.2 Algorithms

In this section, we describe the numerical approach to segment cells in a brightfield image by solving (3.2). The core methodology was adapted from [2]. However, experiments revealed that the main algorithm proposed by Ayed et al. often resulted in over-, or under-segmentation when applied to brightfield images. Thus, Ayed et al.'s main algorithm was used with various approaches such that it can be applied to segment brightfield images with high accuracy. In this paper, we present two algorithms: the Global-Local, and the λ_1-λ_2 approach. Before we explore these novel approaches, the principle steps that are involved to solve (3.2) is outlined:

Main Algorithm
1. Initialize L^n: set $L_p^n = L_p^0 = 1 \ \forall p \in \mathcal{P}$
2. Initialize α_n: set $\alpha_n = \alpha_0$ with $0 < \alpha_0 < 1$
3. Optimize L:
$L^{n+1} = \arg\min_L \lambda \sum_{\{p,q\} \in \mathcal{N}} w_{\{p,q\}} + \sum_{p \in R_0^{L^n}} w_{\{s,p\}} + \sum_{p \in R_1^{L^n}} w_{\{p,t\}}$
4. If $L^n \neq L^{n+1}$,
i) Set $n = n + 1$
ii) Decrease α_n by a factor of ρ: $\alpha_n = \alpha_{n-1}^{\rho}$ for some constant $\rho > 1$
iii) Repeat from step 3
If $L^n = L^{n+1}$,
i) Terminate. L^{n+1} is the final mask of the image.

Table 3.2: Main Algorithm.

L^{n+1} in the optimization step of Table 3.2 can be computed in low-order polynomial time using the max-flow algorithm of Boykov and Kolmogorov in [7].

3.2.1 The Global-Local Technique

The global-local (GL) technique is a two-step segmentation approach that can capture the boundaries of cells in brightfield images effectively. First step of the technique, global segmentation, obtains information about the positioning of the cells. The latter step, local segmentation, captures the fine details of the cells.

The challenging properties that brightfield images possess, such as poor contrast, broken halo, and missing boundaries, makes it difficult to locate the position and the boundary of the cells accurately. Global segmentation takes the original image as a whole, and identifies groups of cells or individual cells into cell blobs (red regions in Figure 3.5). Each blob from global segmentation is identified and is framed into a local region (red dashed boxes of Figure 3.5). For every local region, the Bhattacharyya measure based graph-cut segmentation is applied again. This segmentation process is able to fine-tune the segmentation, separating closely located non-overlapping cells into well separated cells. This step is able to detect the boundary of the cells with low contrast, which tend to be over- or under-segmented during global segmentation, with great detail. The segmentation results from the local step is combined onto the original frame to give the final segmentation of the image. The series of steps involved in the global-local technique can be visualized in Figure 3.5.

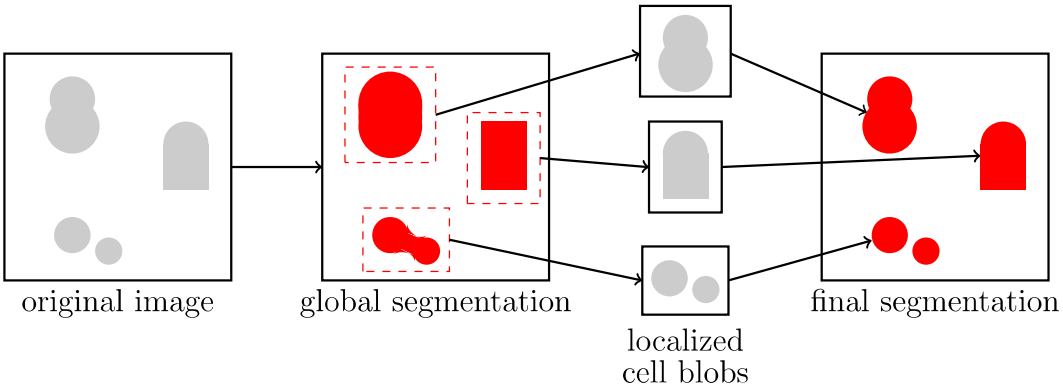


Figure 3.5: A schematic diagram of the global-local technique. Grey objects represent the cells in a given image. Red objects represent the "object" mask. Red-dashed boxes are the local regions to be used for local segmentation.

For each cell blob, one may apply local segmentation recursively. That is, for each new connected blob that appears as a result of "global" segmentation, another "local" segmentation can be performed. One may think of this as a telescopic zoom-in process.

3.2.2 The λ_1-λ_2 Technique

The λ_1-λ_2 technique is also a two-step algorithm. It is based on the role that λ plays in (3.2). Large λ tends to put more emphasis on the boundary term, $B(L) = \sum_{\{p,q\}\in\mathcal{N}} w_{\{p,q\}}$, while small λ values put more emphasis on the regional term. As a result, large λs tend to produce a mask that is well-connected but under-segmented, while small λ values tend to produce disconnected yet over-segmented masks (see Figure 3.6). The λ_1-λ_2 algorithm finds the mask using the Bhattacharyya measure based graph-cut segmentation technique with two different λ values, λ_1 and λ_2. The first λ to be used, λ_1, is the one that tends to over-segment cells, while λ_2 tends to under-segments cells. The segmentation result using λ_1 is used as the initial mask for the segmentation using λ_2 with a fixed number of layers removed (refer to Appendix A for removal of layers).

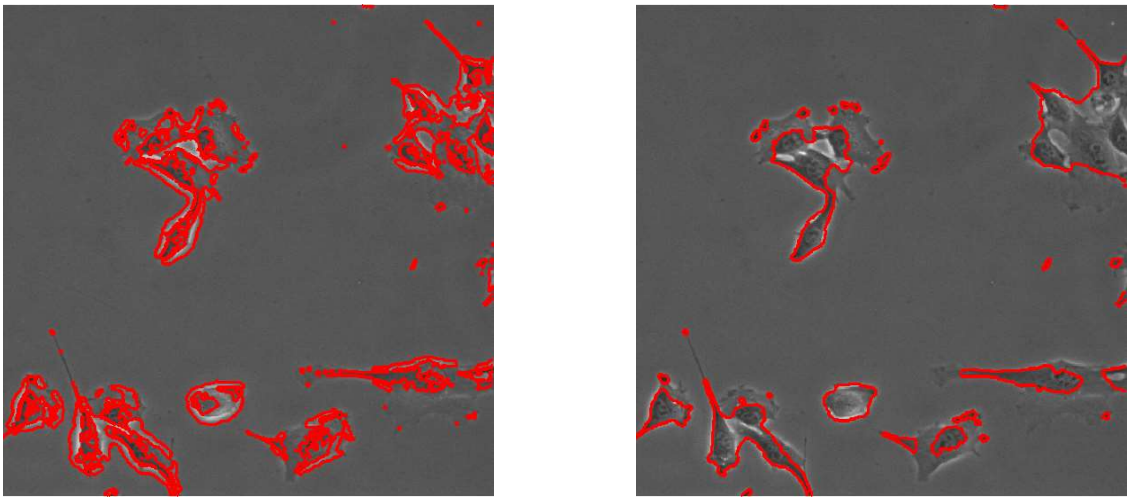


Figure 3.6: Over- and under-segmentation can occur using a different set of λ values. Left: Disconnected over-segmentation occurs with a smaller value of λ, $\lambda = 10^{-3}$. Right: Well-connected under-segmentation occurs with a large value of λ, $\lambda = 10^{-2}$.

It requires trial-and-error to determine which two λ values under- and over-segment cells. However, in a set of images per experiment, specific values of λ_1 and λ_2 tend to share the same property among all other frames.

The λ_1-λ_2 algorithm is outlined in the following table:

λ_1-λ_2 **Algorithm**	
1.	Initialize L^n: set $L_p^n = L_p^0 = 1 \ \forall p \in \mathcal{P}$
2.	Initialize α_n: set $\alpha_n = \alpha_0$ with $0 < \alpha_0 < 1$
3.	Optimize L:
	$L^{n+1} = \arg\min_L \lambda_1 \sum_{\{p,q\} \in \mathcal{N}} w_{\{p,q\}} + \sum_{p \in R_0^{L^n}} w_{\{s,p\}} + \sum_{p \in R_1^{L^n}} w_{\{p,t\}}$
4.	If $L^n \neq L^{n+1}$,
	i) Set $n = n + 1$
	ii) Decrease α_n by a factor of ρ: $\alpha_n = \alpha_{n-1}^{\rho}$ for some constant $\rho > 1$
	iii) Repeat from step 3
	If $L^n = L^{n+1}$,
	i) Proceed to step 5.
5.	Remove some layers of L^{n+1}, and set it to L_p^0
6.	Reinitialize α_n: set $\alpha_n = \alpha_0$ with $0 < \alpha_0 < 1$
7.	Optimize L:
	$L^{n+1} = \arg\min_L \lambda_2 \sum_{\{p,q\} \in \mathcal{N}} w_{\{p,q\}} + \sum_{p \in R_0^{L^n}} w_{\{s,p\}} + \sum_{p \in R_1^{L^n}} w_{\{p,t\}}$
8.	If $L^n \neq L^{n+1}$,
	i) Set $n = n + 1$
	ii) Decrease α_n by a factor of ρ: $\alpha_n = \alpha_{n-1}^{\rho}$ for some constant $\rho > 1$
	iii) Repeat from step 7.
	If $L^n = L^{n+1}$,
	i) Terminate. L^{n+1} is the final mask of the image.

Table 3.3: The λ_1-λ_2 Algorithm.

One may note that steps 6 to 8 are identical to 2 to 4 in the λ_1-λ_2 algorithm. Note that step 5, where the final mask from the segmentation corresponding to optimization for λ_1 is the initial mask of the segmentation process with λ_2, is a crucial step for this method.

Chapter 4

Numerical Results

In this chapter, we present the numerical results of the two methods in sections 3.2.1 and 3.2.2. The algorithms were applied to live C2C12 (muscle) cells that were obtained from the Department of Medicine and Human Genetics at McGill University. In our numerical experiments, we used an 8 neighbourhood-system with the following parameters: $\sigma = 10$, $\alpha_0 = 0.85$, $\rho = 1.1$, and $K = 1000$. It is only ω and λ parameters that varied per experiment. The learned distribution M was obtained by measuring the frequency count of intensities on the border of the image of interest. Images whose borders contained cells had to be manually excluded when obtaining M. The calculations were performed on a MAC with a 2.7GHz processor using MATLAB.

The size of each image frame is 512×512. It was found that graph-cut segmentation on lower resolution images, down-sampled from the original image frame, improved the quality of the segmentation with greater efficiency (see Figure 4.1 and Figure 4.2).

The percentage of the pixels that are correctly classified by the proposed methods can be measured by calculating accuracy in the following way:

$$\text{accuracy} = \frac{|L \cap L_{\text{true}}|}{|L|},$$

where L is the segmentation result, and L_{true} is the ground truth, which can be computed using manual segmentation.

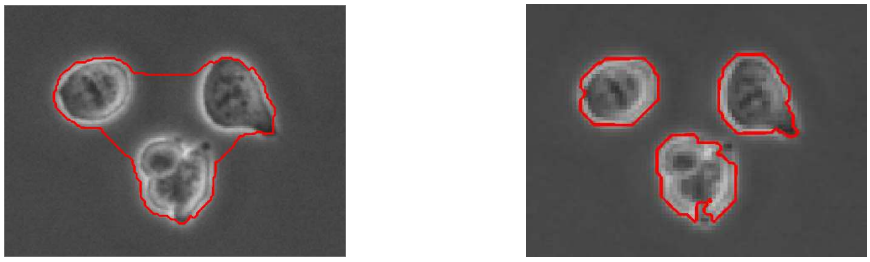


Figure 4.1: Multi-resolution segmentation comparison. Left: Cropped from a 512×512 image. Size: 129×175. Time: 6.31 sec. Parameters: $\omega = 10^{-4}$, $\lambda = 1$. Right: Cropped from a 256×256 image. Size: 65×88. Time: 1.08 sec. Parameters $\omega = 10^{-4}$ and $\lambda = 10$.

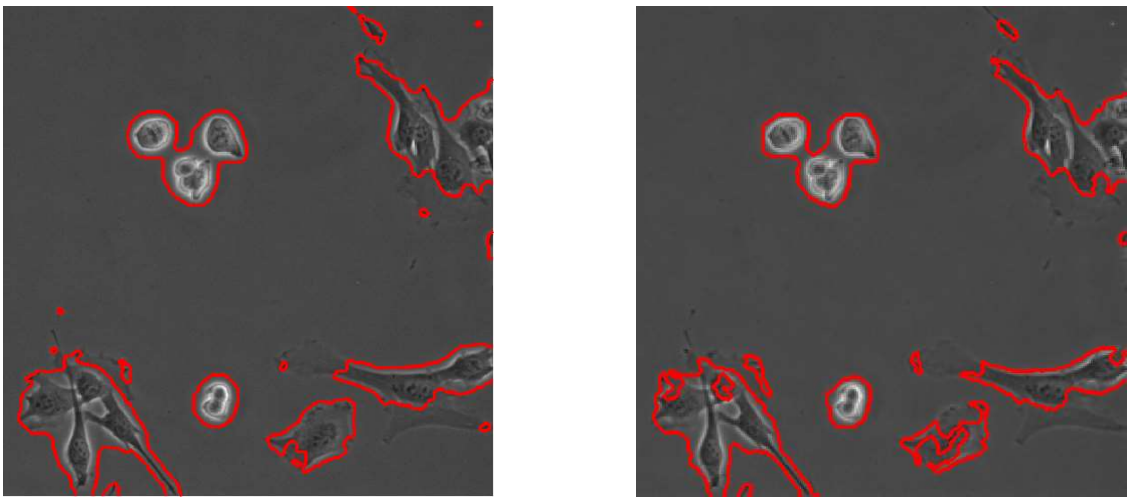


Figure 4.2: Multi-resolution segmentation comparison. Left: Segmentation of 512×512 image took 1377.86 sec. Right: Segmentation of 256×256 took 12.35 sec. Parameters: $\omega = 10^{-4}$, $\lambda = 0.01$ were used for both segmentations.

4.1 Results of the Global-Local Technique

The first two results (Figure 4.3 and Figure 4.4) in this section will be displayed in two rows to demonstrate each step of the GL-scheme in Figure 3.5. The top row will display 3 images: the original image, the global segmentation result, and the final segmentation result in left-to-right sequence. The bottom row will consist of results from local segmentation.

Table 4.1 and Table 4.2 will give information on the parameters that were used to obtain images in Figure 4.3 and Figure 4.4, respectively.

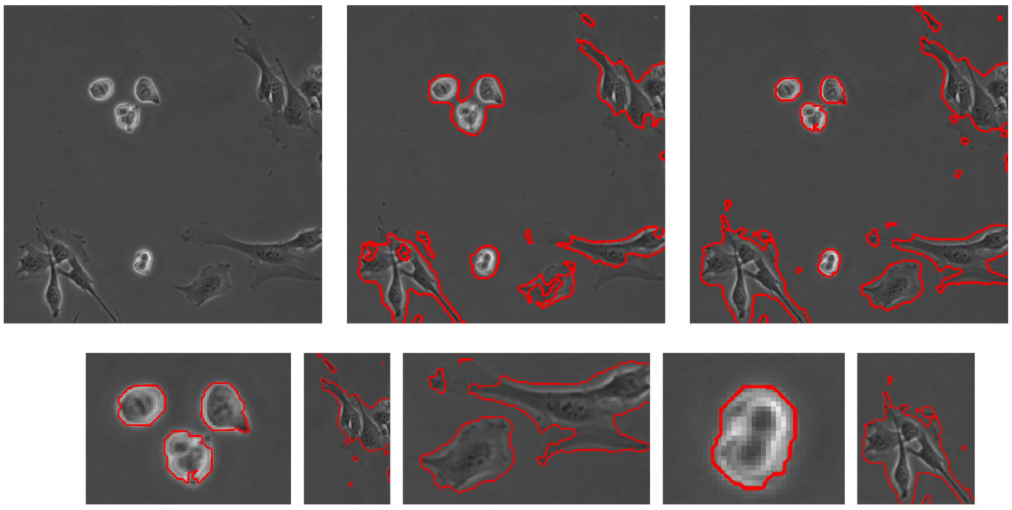


Figure 4.3: Top-left: original image. Top-middle: result of global segmentation. Top-right: result of final segmentation. Bottom-left to bottom-right images are local segmentation results of blobs $1, \ldots, 5$.

	Global	Blob 1	Blob 2	Blob 3	Blob 4	Blob 5
Size	256×256	65×88	156×89	78×127	31×37	117×91
ω	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}	10^{-4}
λ	10^{-2}	10	0.1	0.1	10	0.1
Time	12.63 sec	1.25 sec	4.09 sec	3.39 sec	1.60 sec	3.40 sec

Table 4.1: Parameters and the running times of each segmentation result in Figure 4.3.

In Figure 4.3, global segmentation (top-middle) is able to define five major regions of the original image (top-left). However, the cell boundaries are not very precise, and nearby cells are identified as one big blob. In local segmentation, the trio of cells (left-most bottom) that were identified as a blob because they were close together, is identified as three individual cells. The final mask has an accuracy of 92.89%.

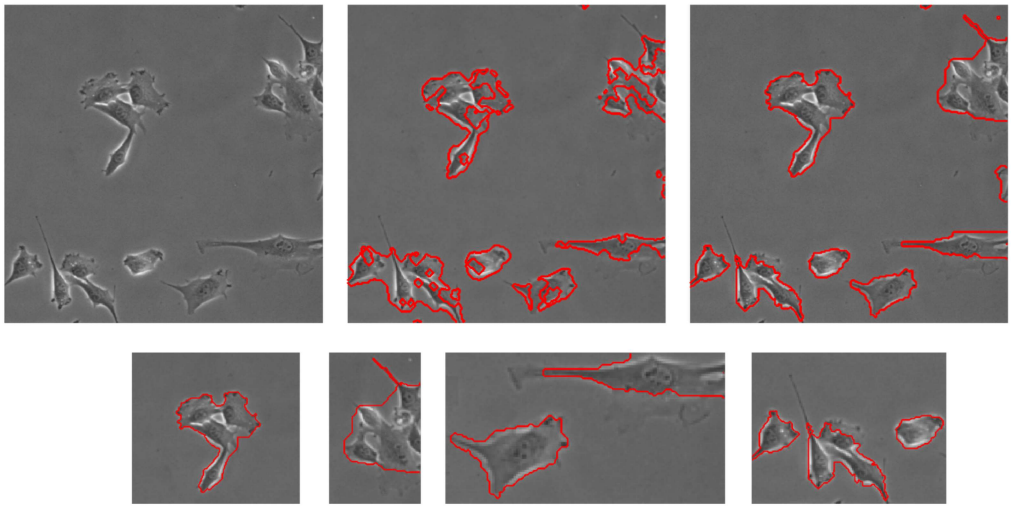


Figure 4.4: Top-left: original image. Top-middle: result of global segmentation. Top-right: result of final segmentation. Bottom-left to bottom-right images are local segmentation results of blobs $1, \ldots, 4$.

	Global	Blob 1	Blob 2	Blob 3	Blob 4
Size	256×256	129×143	111×67	71×131	101×130
ω	10^{-4}	10^{-2}	10^{-4}	$\sigma = 10^{-4}$	$\sigma = 10^{-3}$
λ	10^{-2}	10^{-2}	$\lambda = 1$	$\lambda = 1$	$\lambda = 0.1$
Time	14.58 sec	3.50 sec	4.08 sec	6.44 sec	4.01 sec

Table 4.2: Parameters and the running times of each segmentation result in Figure 4.4.

In Figure 4.4, global segmentation is able to identify four blobs via graph-cut segmentation using the Bhattacharyya measure. Again, the boundaries obtained in global segmentation are not very precise, especially those cells with low contrast. However, in local segmentation, the boundaries of poorly contrasted cells are well defined (bottom-mid-right). The final mask has an accuracy of 92.55%.

Table 4.3 displays a sequence of frames in a single experiment to track the activity of the cells by segmenting each frame individually.

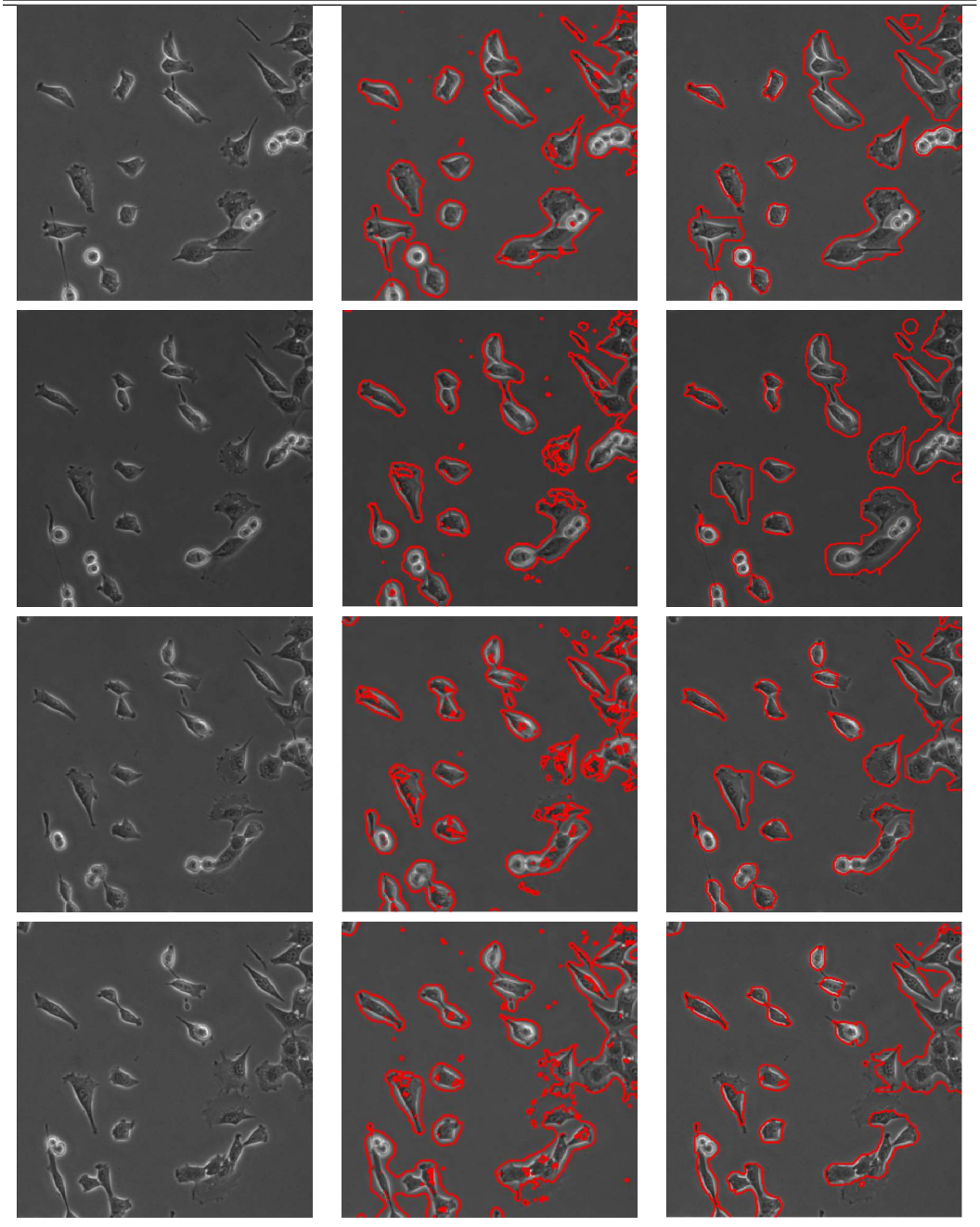
Original Image	Global Segmentation	Final Segmentation
		

Table 4.3: Sequence of frames segmented using the GL-technique.

4.2 Results of the λ_1-λ_2 Technique

Unlike the global-local method, which begins each graph-cut segmentation with a blank mask in both global- and local-segmentation steps, the λ_1-λ_2 technique uses the final segmentation result of λ_1-segmentation as the initial mask of the λ_2-segmentation. Through experiments, it was found that a more desired solution is achieved when $\lambda_1 < \lambda_2$. That is, a more accurate mask is determined when the first graph-cut segmentation emphasizes the regional terms, the labelling of the pixels, then on the boundary term to balance the two in (3.2).

The results of λ_1 segmentation and λ_2 segmentation are displayed in Table 4.5, and the parameters that were used, as well as its run times and the accuracy of its results can be found in Table 4.4. Note that the images for λ_1-λ_2 were not down-sampled because we were able to attain relatively accurate results without down-sampling. Thus, running times will be incomparably longer than those from the global-local technique.

	Image 1	Image 2	Image 3	Image 4
ω	10^{-4}	10^{-4}	10^{-4}	10^{-4}
λ_1	10^{-3}	10^{-3}	10^{-3}	10^{-3}
λ_2	10^{-2}	10^{-2}	10^{-2}	10^{-2}
Time	212.98 sec	302.81 sec	295.01 sec	868.19 sec
Accuracy	92.04%	91.00%	90.64%	89.61%

Table 4.4: Parameters used to attain results in Table 4.5.

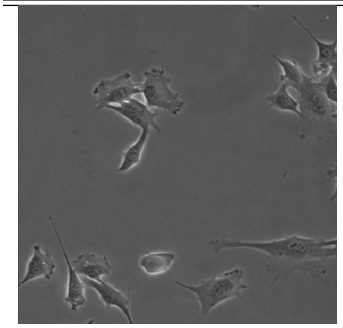
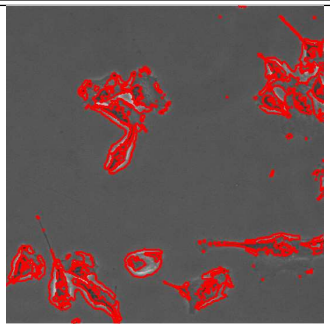
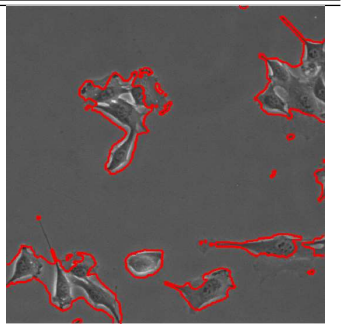
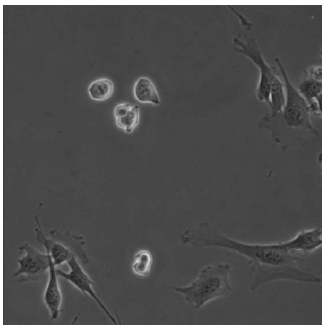
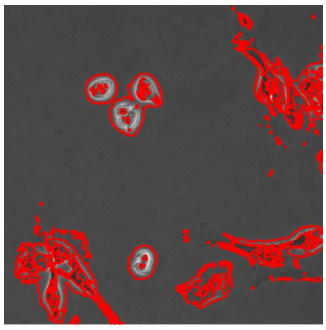
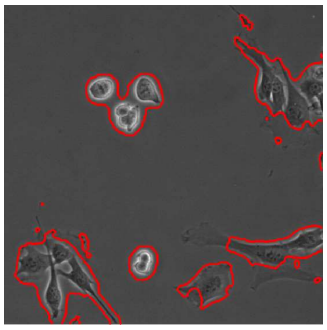
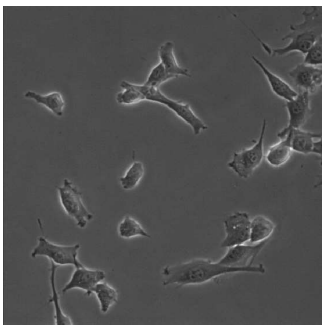
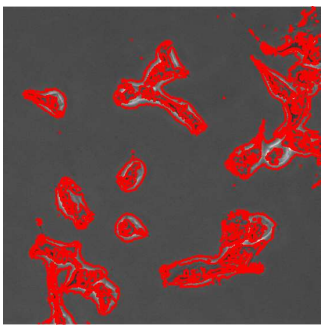
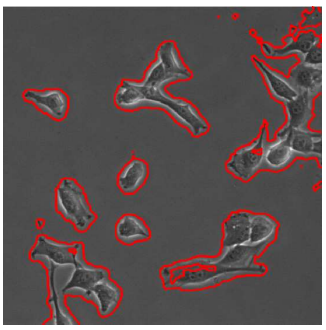
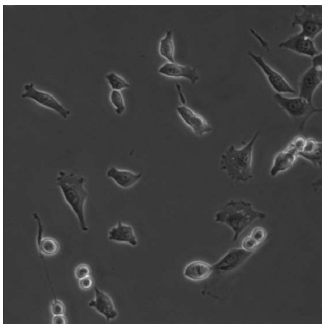
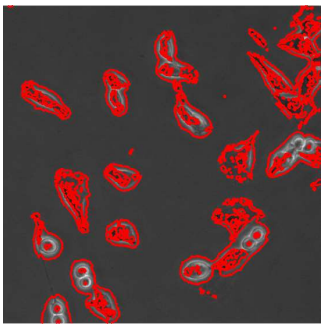
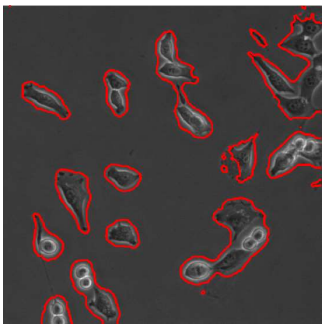
Original Image	λ_1-segmentation	λ_2-segmentation
		
		
		
		

Table 4.5: Sequence of frames segmented using the λ_1-λ_2 technique.

4.3 Comparison of the two methods

In this section, we compare between the two methods proposed through qualitative and quantitative analysis in Table 4.6. Both algorithms are able to capture cells of different sizes and shapes but some cells with low contrast are over- and/or under-segmented. Although the global-local technique is able to capture fine details on well-separated cells, λ_1-λ_2 technique is able to segment overlapping cells with greater precision. The λ_1-λ_2 algorithm takes longer than the global-local algorithm by several factors. However, as mentioned earlier, this is because the images were not downsized in the λ_1-λ_2 experiments, as they were for the global-local algorithm.

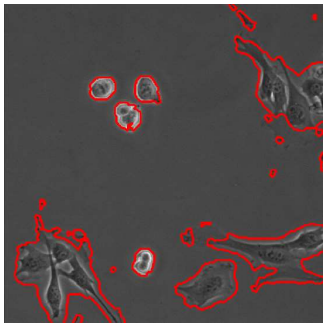
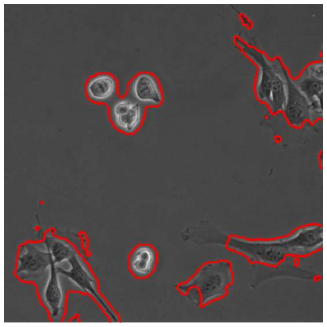
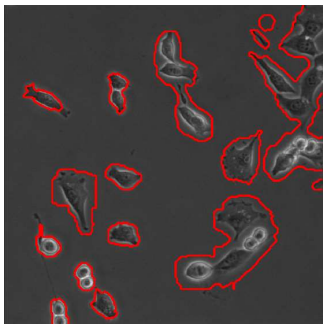
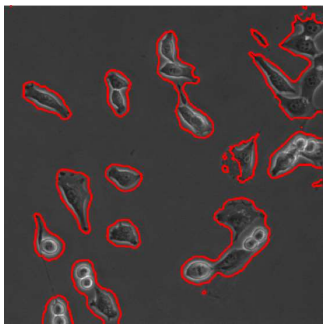
GL Results	λ_1-λ_2 Results	Quantitative Comparison
		$\text{accuracy}_{\text{GL}} = 92.89\%$ $\text{time}_{\text{GL}} = 26.36$ sec $\text{accuracy}_{\lambda_1,\lambda_2} = 91.00\%$ $\text{time}_{\lambda_1,\lambda_2} = 302.81$ sec
		$\text{accuracy}_{\text{GL}} = 90.50\%$ $\text{time}_{\text{GL}} = 28.22$ sec $\text{accuracy}_{\lambda_1,\lambda_2} = 89.61\%$ $\text{time}_{\lambda_1,\lambda_2} = 868.19$ sec

Table 4.6: Segmentation method comparisons of the global-local (GL) and the λ_1-λ_2 techniques.

By comparing Tables 4.1 and 4.2 with 4.4, one can see that the parameters for the global-local method requires specific parameters per case to obtain a good result, while

λ_1-λ_2 algorithm is able to segment images with consistent accuracy with a fixed set of parameters.

4.4 Remarks

The segmentation results of both techniques can be improved by providing more information to its initial mask. That is, instead of providing a mask, which is entirely labelled as the background, providing some hard constraints prior to segmentation can improve the results. This can be done by manually identifying some cells as "object," using the segmentation result from previous frames as initial of current frame, or by using the result of the global-local techniques as the initial mask for the λ_1-λ_2 algorithm and vice versa. This was not shown in preceding sections to emphasize that the two algorithms presented in this paper are capable of segmenting brightfield images without prior information with high accuracy.

Chapter 5

Conclusion

This paper has presented a solution to one of the challenges of segmenting cells in brightfield microscopy images. In particular, the two different techniques were introduced using graph-cut segmentation via the Bhattacharyya measure.

The first model, referred to as the global-local technique, is a multi-scale procedure that involves determination of the location of the cells to fine-tune the segmentation of each cell or collection of cells that are close together. The second model, λ_1-λ_2 technique, uses two different λ values to put different emphasis on the energy function that we wish to minimize.

Both algorithms were able to segment cells in images that contained multiple cells with high accuracy, and required very little input from the user. Information from previous frames were found to be unnecessary, although they can be helped to obtain an even more accurate solution.

Possible future work includes further investigation to identify individual cells that overlap one another.

APPENDIX

Appendix A

Removing a layer of a mask

Suppose a binary mask is given as in Figure A.1. To remove a layer, consider pixel (i,j), and look at its surrounding 8-pixels. If any one of the eight pixels do not have the same black (or white) label as pixel (i,j), then pixel (i,j) must change its label from black to white (or white to black). This is applied to all pixels in the mask for a single layer of removal. In Figure A.1, since all eight surrounding pixels of (i,j) are of the same label as (i,j), black, (i,j) maintains its black label. However, pixel (i-1,j-1), for example, does not have the same black label as four of its 8-neighbouring pixels. Namely, (i-1,j-2), (i-2,j-2), (i-2,j-1), and (i-2,j). Therefore, in a single layer removal stage, the (i-1,j-1) is converted to its opposite label, white. Similarly, pixels (i,j-1) (i-1,j-1), (i-1,j), (i-1,j+1), (i,j-1), (i,j+1), (i+1,j), and (i+1,j+1) are also converted in a single layer removal stage. To remove n-layers, $(2n+1)^2 - 1$ neighbouring pixels must be considered.

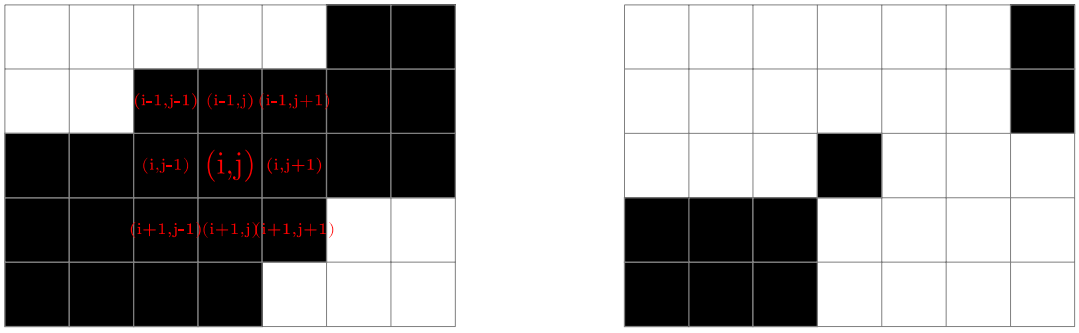


Figure A.1: Removing a layer of a mask. Left: Original Mask. Right: Resulting mask with 1 layer removed.

References

[1] R. Ali, M. Gooding, T. Szilagyi, B. Vojnovic, M. Christlieb, and M. Brady. Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images. *Machine Vision and Applications*, 23:607–621, 2012.

[2] I. Ayed, H.M. Chen, K. Punithakumar, I. Ross, and S. Li. Graph Cut Segmentation with a Global Constraint: recovering region distribution via a bound of the Bhattacharyya measure. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–7, 2010.

[3] I. Ayed, H.M. Chen, K. Punithakumar, I. Ross, and S. Li. Max-flow segmentation of the left ventricle by recovering subject-specific distributions via a bound of the Bhattacharyya measure. *Medical Image Analysis*, 16:87–100, 2012.

[4] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific and Dynamic Ideas, 1997.

[5] Y. Boykov and G. Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, 2006.

[6] Y. Boykov and M. Jolly. Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images. In *International Conference on Computer Vision*, volume 1, pages 105–112, 2001.

[7] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. In *IEEE Transactions on PAMI*, volume 26, pages 1124–1137, 2004.

[8] L. Bradbury. Segmentation of Bright-field Cell Images. Master's thesis, University of Waterloo, 2009.

[9] L. Ford and D. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[10] A. Goldberg and R. Tarjan. A new approach to the maximum-flow problem, 1988.

[11] M. Hariri. Reconstruction of Incomplete Cell Paths Through Three-Dimensional Segmentation. Master's thesis, University of Waterloo, 2011.

[12] S. Inoue. *Video Microscopy*. Springer, 1986.

[13] A. Korzynska and M. Iwanowski. Multistage Morphological segmentation of bright-field and fluorescent microscopy images. *Opto-Electronics Review*, 20(2):174–186, 2012.

[14] A. Korzynska, W. Strojny, A. Hoppe, D. Wertheim, and P. Hoser. Segmentation of microscope images of living cells. *Pattern Analysis and Applications*, 10:301–319, 2007.

[15] K. Li and T. Kanade. Cell Population Tracking and Lineage Construction Using Multiple-Model Dynamics Filters and Spatiotemporal Optimization. In *Proceedings of the 2nd International Workshop on Microscopic Image Analysis with Applications in Biology (MIAAB)*, September 2007.

[16] A. Massoudi, A. Sowmya, K. Mele, and Semenovich D. Employing Temporal Information for Cell Segmentation Using Max-flow/Min-cut in Phase-Contrast Video Microscopy. In *33rd Annual International Conference of the IEEE EMBS*, 2011.

[17] M. Wang, X. Zhou, F. Li, J. Huckins, and R. King. Novel cell segmentation and online SVM for cell cycle phase identification in automated microscopy. *Bioinformatics*, 24(1):94–101, 2008.

[18] K. Wu, D. Gauthier, and M Levine. Live cell image segmentation. In *IEEE Transactions on Biomedical Engineering*, volume 42, pages 1–12, 1995.