

Automated Segmentation of Cellular Images using an Effective Region Force

by

Khadeejah Mohiuddin

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, 2017

© Khadeejah Mohiuddin. Public 2017

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

Understanding the behaviour of cells is an important problem for biologists. Significant research has been done to facilitate this by automating the segmentation of microscopic cellular images. Bright-field images of cells prove to be particularly difficult to segment due to features such as low contrast, missing boundaries and broken halos. In this paper, we present two algorithms for automated segmentation of cellular images. These algorithms are based on a graph-partitioning approach where each pixel is modelled as a node of a weighted graph. The method combines an effective Region Force with the Laplacian and the Total Variation boundary forces, respectively, to give the two models. This region force can be interpreted as a conditional probability of a pixel belonging to a certain class (cell or background) given a small set of already labelled pixels. For practicality, we use a small set of only background pixels from the border of cell images as the labelled set. Both algorithms are tested on bright-field images to give good results. Due to faster performance, the Laplacian-based algorithm is also tested on a variety of other datasets including fluorescent images, phase-contrast images and 2- and 3-D simulated images. The results show that the algorithm performs well and consistently across a range of various cell image features such as the cell shape, size, contrast and noise levels.

Acknowledgements

I would like to thank my supervisor, Prof. Justin W. L. Wan, for his valuable advice and guidance throughout my graduate study at the University of Waterloo. I would also like to thank my family for their constant support and my colleagues in the Computational Math program who made studying all the more exciting.

Dedication

This paper is dedicated to my parents. Their love and support is my driving force.

Table of Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Cellular Images	1
1.1.1 Bright-field Microscopy	2
1.1.2 Fluorescent Microscopy	2
1.1.3 Phase-Contrast Microscopy	3
1.2 Cell Image Segmentation	3
2 Background	5
2.1 Cell Image Segmentation	5
2.2 Graphical Model	5
2.3 Graph Operators	6
2.4 Graph Partitioning Problem	7
2.5 The Chan-Vese Model	11
3 Methodology	13
3.1 Boundary and Region Forces	13
3.1.1 Boundary Force	14

3.1.2	Region Force	14
3.2	Algorithms for Graph Partitioning with a Region Force	16
3.2.1	Laplacian-Based	16
3.2.2	TV-Based	17
3.3	Cellular Image Considerations	18
4	Numerical Results	20
4.1	Bright-Field Images	20
4.2	ISBI 2013 Cell Tracking Challenge Datasets	21
4.3	Cell Tracking Challenge 2017 Datasets	22
4.3.1	2-D Images	30
4.3.2	3-D Images	31
5	Conclusion	40
	References	42

List of Tables

4.1	Time Comparison of Segmentation on Bright-field Image	21
4.2	ISBI 2013 Cell Tracking Challenge scores and LapRF	30
4.3	Time Comparison of Segmentation on 3-D Image	31

List of Figures

1.1	Comparison of Bright-field and Fluorescent Image	2
2.1	Two-Moons Dataset	12
4.1	Bright-field Image	22
4.2	Segmentation Results on Bright-field Image	23
4.3	Segmentation on N2DH-SIM image – video 01	24
4.4	Segmentation on N2DH-SIM image – video 02	25
4.5	Segmentation on N2DH-SIM image – video 03	26
4.6	Segmentation on N2DH-SIM image – video 04	27
4.7	Segmentation on N2DH-SIM image – video 05	28
4.8	Segmentation on N2DH-SIM image – video 06	29
4.9	Segmentation on Fluo-C2DL-MSK image	32
4.10	Segmentation on Fluo-N2DH-GOWT1 image	33
4.11	Segmentation on Fluo-N2DL-HeLa image	34
4.12	Segmentation on PhC-C2DH-U373 image	35
4.13	Segmentation on PhC-C2DL-PSC image	36
4.14	3-D Image Slices	37
4.15	3-D Image Segmentation Results	38
4.16	3-D Image Single Slice Segmentation	39

Chapter 1

Introduction

Image segmentation is a well-developed area in computer vision. It is the process of automatically identifying the location of objects in an image. The objects could be as simple as the foreground and background or more complex such as the eyes and nose on a picture of a face. Successfully locating the objects on an image is usually the first step to further analysis and interpretation of the image. In our research essay, we focus on the segmentation of cellular images. The goal of cell image segmentation is to locate the cells from the background medium they are contained in.

1.1 Cellular Images

Cells are the fundamental units of life and understanding their behaviour can give valuable information to biologists. The study of cells is typically done on time-lapse microscopy data sets that contain a large number of image frames. This makes manual analysis very time consuming and error prone, especially in 3-D [17]. Therefore, automatic segmentation of cellular images becomes a necessary first stage in analyzing the behaviour of cells. There are three common cellular imaging techniques used; bright-field, fluorescent, and phase-contrast microscopy, and they all have unique properties that make them more or less suitable for segmentation. A comparison of a bright-field and fluorescent image of the same sample of cells is given in Figure 1.1.

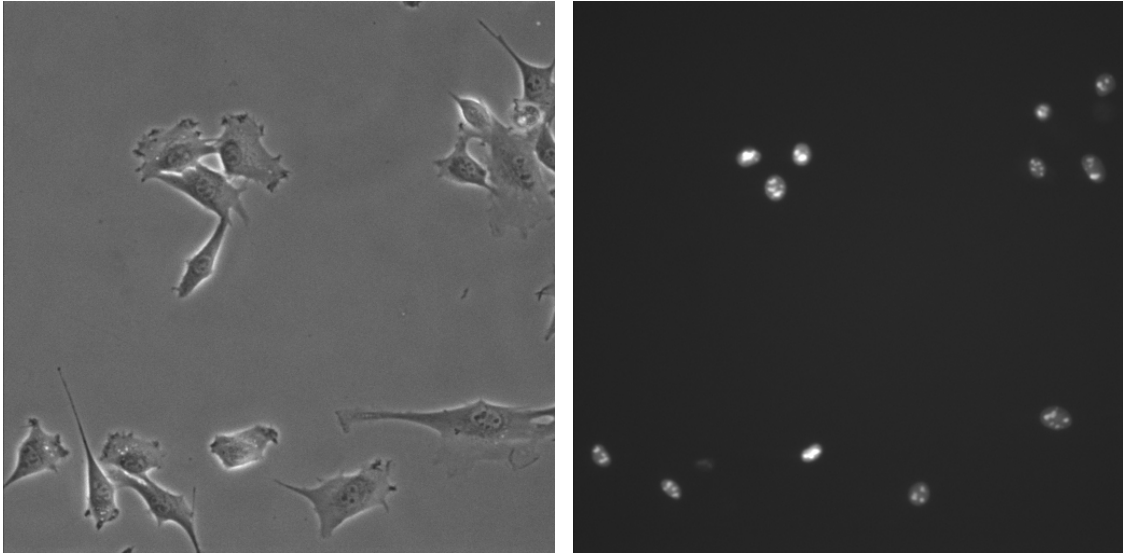


Figure 1.1: A comparison of a bright-field (left) and fluorescent (right) image of the same sample of cells.

1.1.1 Bright-field Microscopy

Bright-field microscopy is one of the most common microscopy techniques. It involves transmitting light from the bottom of the sample of cells and viewing the light that emerges from the top. Because different parts of the cell and background absorb light to different degrees, an image of bright and dark areas is formed. There is usually low contrast between the background and the centre of the cell, and the most discernible areas are a dark belt and a bright “halo” surrounding the border of a cell. This is caused by light converging through the upright cell membranes. However, some cell areas can be flattened along with their membranes which results in an inconsistent border around the cell and a broken halo. These inconsistent boundaries and poor contrast are some of the challenges of segmenting bright-field images [18].

1.1.2 Fluorescent Microscopy

Fluorescent microscopy involves the staining of cells which fluoresce when light of a particular wavelength is emitted through them. This creates a high contrast between the stained part of cells and the background– which makes it a good choice for segmentation. However,

in most cases, only certain organelles such as the nucleus is stained and captured. This makes it unsuitable for tasks where the entire cell is to be analyzed. Another issue with fluorescent imaging is that the stain may influence the way a cell moves, grows or divides so it might not capture the true behaviour of cells [19].

1.1.3 Phase-Contrast Microscopy

Phase-Contrast microscopy is based on the principle that light passing through an area of different refractive index undergoes a phase shift. This change in phase is then translated into a variation in brightness. Unlike the fluorescent method, cell samples for the phase-contrast images need not be stained and therefore complete cells can be captured without altering their true behaviour. However, while the phase-contrast method is better than bright-field in terms of contrast, it still produces images with artifacts such a halo and a shadow around the cells [21].

1.2 Cell Image Segmentation

There has been considerable research done on the segmentation of cellular images. The high-contrast feature of fluorescent imaging makes it a particularly popular choice for experiment. Nevertheless, studies of bright-field image segmentation, although fewer, have also been done. In [19], Bradbury et al. propose a spectral-k-means clustering algorithm for segmentation of bright-field cell images. However, the algorithm is made to locate the boundaries of a cell images with only one cell. When multiple cells are present, it requires prior information about the location of each cell. Another approach based on using active contour methods for bright-field image segmentation is given by Tscherepanow et al. in [20]. The method is successful but is only tested on one type of cell. In [22], Kang et al. use a graph-partitioning approach combined with a global-local method that fine-tunes the result by applying segmentation twice. First on the entire image to obtain large “blobs” of groups of cells, and then again on each blob to obtain a refined segmentation of individual cells. This means it requires extra information about of the location of each blob. Being tested on one type of cell, it also leaves some doubt as to how consistently it performs on other datasets. Finally, a unique cell image segmentation algorithm based on a game theoretic approach is given by Dimock et al. in [23]. It is tested on a variety of 2-D datasets to give good results.

The approach to be presented in this paper is based on a graph-partitioning problem

which combines a local and a global metric for segmentation. The goal is to segment an image with multiple cells without user input about the location of each cell or group of cells. To test consistency and robustness, it will be applied on bright-field images as well as fluorescent, phase-contrast, and 2- and 3-D simulated images.

In Chapter 2 we give some useful background knowledge about the models and algorithms that are presented in Chapter 3. The numerical results of applying those models on various cell images are given in Chapter 4 and the conclusion in Chapter 5.

Chapter 2

Background

The algorithms presented in this paper rely on a graphical model to represent the image and its characteristics. In this chapter we first define our objective that is image segmentation, and then move on to give a basic understanding of our graphical model, some useful graph operators, and the graph partitioning problem. Finally, we present the celebrated Chan-Vese Model.

2.1 Cell Image Segmentation

Pixels, which are very small boxes of different intensities, are the building blocks of our cell images. Many of them put together can be recognized by the human eye as a picture of cells. Cell image segmentation is the process of automatically identifying which of those pixels make up the cells and which of those pixels make up the background. This means we have two *clusters*; cell and background, that pixels can be labelled as.

The pixel intensity and location can be used to construct a data vector corresponding to each pixel. These data vectors can then be used to build our graphical model.

2.2 Graphical Model

A weighted undirected graph $G = (V, E, w)$ consists of a set of nodes or vertices V , edges E that connect those vertices, and a weight function $w : E \rightarrow R_+$ defined on those edges.

We say that a graph is undirected if for a pair of nodes $x_i, x_j \in V$, the edges $\{x_i, x_j\}$ and $\{x_j, x_i\}$ are considered to be the same. The nodes in the graph can correspond to the pixel data vectors and the weight function can be defined in such a way that it assigns larger weights to edges between nodes that are similar and smaller weights to edges between nodes that are different. This means that for every edge $\{x_i, x_j\} \in E$, the weight $w_{ij} = w(x_i, x_j)$ should measure how “close” the nodes x_i and x_j are.

There are many popular choices of weight functions to measure the similarity between a pair of nodes, such as the Radial Basis Function (RBF) [1]

$$w(x_i, x_j) = \exp\left(\frac{-d(x_i, x_j)^2}{2\omega}\right), \quad (2.1)$$

where d is a distance metric and ω is a constant, the Zelnik-Manor and Perona (ZMP) weight function [2]

$$w(x_i, x_j) = \exp\left(\frac{-d(x_i, x_j)^2}{\sqrt{\sigma(x_i)\sigma(x_j)}}\right), \quad (2.2)$$

where σ is the local variance, and the Cosine Similarity [3]

$$w(x_i, x_j) = \cos(x_i, x_j) = \frac{\langle x_i, x_j \rangle}{\|x_i\| \|x_j\|}, \quad (2.3)$$

which is particularly useful in natural language processing.

In practice, this weight is not assigned between all pairs of nodes. The weight matrix $W = (w_{ij})$, also called *affinity* matrix, can instead be constructed by a k -Nearest Neighbour (k -NN) search. That is, for each node, we assign this weight only on the edges connecting it to its closest k number of neighbours. All other edges connected to this node receive a zero weight. Therefore, it can be assumed that the graph is sparse and that each node has at most k neighbours.

2.3 Graph Operators

The affinity matrix W is very important and gives rise to many useful graph operators [4]. Suppose D is a diagonal matrix with the entries on the main diagonal equal to the sum

across the rows of W . Then the *unnormalized graph Laplacian* is given by

$$L = D - W, \quad (2.4)$$

which is a symmetric matrix since both D and W are symmetric. The *Normalized Affinity Matrix* is defined by

$$\hat{W} = D^{-1/2}WD^{-1/2}, \quad (2.5)$$

which is also symmetric. Similarly, the *Normalized graph Laplacian* is given by

$$L_s = D^{-1/2}LD^{-1/2} = I - \hat{W}. \quad (2.6)$$

If $u \in L^2(V)$ is a function defined on the node set of a graph, then the *gradient* operator $\nabla : L^2(V) \rightarrow L^2(V, L^2(V))$ is given by [5, 8]

$$\nabla u(x_i)(x_j) = w_{ij}(u(x_j) - u(x_i)). \quad (2.7)$$

So $\nabla u(x_i)$ is a graph function for each $x_i \in V$. Since G is a sparse graph with at most k neighbours for each node x_i , $\nabla u(x_i)$ can be seen as a vector with at most k non zero elements which are in the neighbourhood of x_j . That is,

$$\nabla u(x_i) = (w_{ij}(u(x_j) - u(x_i)))_{x_j \in \mathcal{N}(x_i)}, \quad (2.8)$$

where $\mathcal{N}(x_i)$ is the set of neighbours of x_i . Furthermore, the *divergence* operator $div : L^2(V, L^2(V)) \rightarrow L^2(V)$ is defined by

$$div(f)(x_i) = \sum_{x_j \in \mathcal{N}(x_i)} w_{ij}(f(x_j)(x_i) - f(x_i)(x_j)). \quad (2.9)$$

2.4 Graph Partitioning Problem

The objective of cell image segmentation is to label each pixel of a cell image as either cell pixel or background pixel. If each pixel corresponds to a node, then partitioning the node set V of the graph into two disjoint sets is equivalent to labelling each pixel as either cell or background. However, image segmentation and graph partitioning is not limited to only two clusters so we will present the generalized case of K clusters. Graph partitioning is the process of dividing the node set of a graph into disjoint subsets or clusters by removing some edges in such a way that there are no edges that connect a node from one subset to another. A *cut* C is the set of edges that were removed in making this partition. The cost

of a cut $|C|$ is the sum of the weights of the edges in C . Therefore, to partition a graph into K clusters V_1, \dots, V_K , the cost function is given by

$$|C| = \sum_{k=1}^K \sum_{x_i \in V_k, x_j \in V/V_k} w_{ij}. \quad (2.10)$$

In Section 2.2, we saw that edges between nodes that are closer together have larger weights than those that are further apart. So, to partition the graph into nodes of different characteristics or classes, the sum of the weights of the edges that are removed, or the cost of the cut, should be minimized to obtain an optimal partition. This gives the graph partitioning problem

$$\{V_k\} = \operatorname{argmin}_{\{V_k\}} \sum_{k=1}^K \sum_{x_i \in V_k, x_j \in V/V_k} w_{ij}. \quad (2.11)$$

This term is known as the boundary force in the context of image segmentation. It works to separate nodes that are sufficiently further apart into different clusters and keep nodes that are sufficiently close together in the same cluster. Therefore, it can be said that boundary forces penalize the discontinuity between neighbouring nodes. Equation (2.11) can be reformulated in terms of a partition matrix $\Phi = (\phi_{ik})$ defined by

$$\phi_{ik} = \begin{cases} 1 & \text{if } x_i \in V_k \\ 0 & \text{otherwise.} \end{cases}$$

If we have n number of nodes then this matrix has dimensions $n \times K$. The partition matrix is a way of representing which cluster each node in the graph belongs to. That is, column k of row i of the partition matrix takes on the value of 1 if x_i belongs to cluster k and all other columns of this row take on the value 0. It can also be seen that if nodes x_i and x_j belong to the same cluster k , then $\phi_{ik} = \phi_{jk} = 1$ and $\phi_{ik} - \phi_{jk} = 0$. If they belong to different clusters, $\phi_{ik} - \phi_{jk} = 1$ or -1 . Since a cut edge between two nodes puts those nodes into different clusters, we can simplify (2.11) to [9, 10]:

$$\Phi = \operatorname{argmin}_{\Phi} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} w_{ij} |\phi_{ik} - \phi_{jk}| \quad \text{or} \quad (2.12)$$

$$\Phi = \operatorname{argmin}_{\Phi} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} w_{ij} (\phi_{ik} - \phi_{jk})^2. \quad (2.13)$$

This is because the terms $|\phi_{ik} - \phi_{jk}|$ and $(\phi_{ik} - \phi_{jk})^2$ simplify to zero if two nodes belong to the same cluster or to one if they belong to different clusters. The same can be said for $(\phi_{ik} - \phi_{jk})^p$ where p is any multiple of 2. However, we only consider the two cases defined above because it will be seen further on that they can be simplified into forms that allow us to effectively formulate and solve the problems. Moving on, since each node can only belong to one cluster, each row of Φ must add up to one. This gives the constraint

$$\Phi \mathbf{1} = \mathbf{1}. \quad (2.14)$$

Simplifying equation (2.12) gives us:

$$\begin{aligned} \Phi &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} w_{ij} |\phi_{ik} - \phi_{jk}| \\ &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} |w_{ij} ((\Phi_k)_i - (\Phi_k)_j)| \\ &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \|\nabla \Phi_k\|_1, \end{aligned} \quad (2.15)$$

where Φ_k is the k -th column of Φ . When combined with the constraint (2.14) it gives us the *Total Variation (TV)* formulation for graph-partitioning,

$$\begin{aligned} \Phi &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \|\nabla \Phi_k\|_1 \\ \text{s.t. } \Phi \mathbf{1} &= \mathbf{1}. \end{aligned} \quad (2.16)$$

The Total Variation of a function is generally defined as the integral of the absolute gradient of the function. We will also simplify equation (2.13). Let d_i denote the elements on the

diagonal of matrix D , then

$$\begin{aligned}
\Phi &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} w_{ij} (\phi_{ik} - \phi_{jk})^2 \\
&= \operatorname{argmin}_{\Phi} \frac{1}{2} \sum_{k=1}^K \sum_{\{x_i, x_j\} \in E} w_{ij} ((\Phi_k)_i - (\Phi_k)_j)^2 \\
&= \operatorname{argmin}_{\Phi} \frac{1}{2} \sum_{k=1}^K \sum_{i,j=1}^n \left(w_{ij} (\Phi_k)_i^2 - 2(\Phi_k)_i (\Phi_k)_j w_{ij} + w_{ij} (\Phi_k)_j^2 \right) \\
&= \operatorname{argmin}_{\Phi} \frac{1}{2} \sum_{k=1}^K \left(\sum_{i=1}^n d_i (\Phi_k)_i^2 - 2 \sum_{i,j=1}^n (\Phi_k)_i (\Phi_k)_j w_{ij} + \sum_{j=1}^n d_j (\Phi_k)_j^2 \right) \\
&= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \left(\sum_{i=1}^n d_i (\Phi_k)_i^2 - \sum_{i,j=1}^n (\Phi_k)_i (\Phi_k)_j w_{ij} \right) \\
&= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \left(\Phi_k^T D \Phi_k - \Phi_k^T W \Phi_k \right) \\
&= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \Phi_k^T L \Phi_k. \tag{2.17}
\end{aligned}$$

Similarly, combining it with the constraint gives us the *Laplacian-based* formulation for graph-partitioning,

$$\begin{aligned}
\Phi &= \operatorname{argmin}_{\Phi} \sum_{k=1}^K \Phi_k^T L \Phi_k \\
\text{s.t. } \Phi \mathbf{1} &= \mathbf{1}. \tag{2.18}
\end{aligned}$$

2.5 The Chan-Vese Model

The well known Chan-Vese Model [6] for graph partitioning combines the TV boundary force defined in (2.16) with another term called the region force. It is defined as [7]

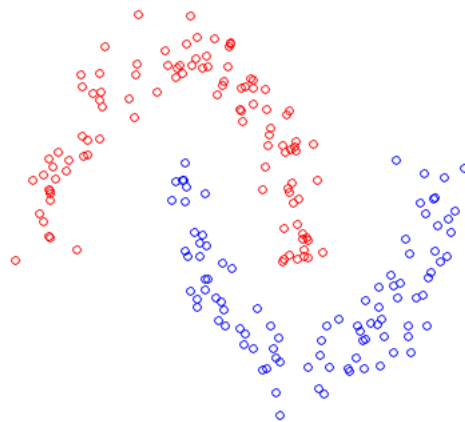
$$\begin{aligned} \Phi &= \arg \min_{\phi_{ik} \in [0,1]} \sum_{k=1}^K \sum_{x_i \in V} (g_k(x_i) \|\nabla(\Phi_k)_i\| + \mu \phi_{ik} \|x_i - c_k\|^2) \\ \text{s.t. } \Phi 1 &= 1, \end{aligned} \tag{2.19}$$

where μ is a constant and $g_k(x_i)$ is the edge detector function which, for the application considered in this paper, can be simplified to 1. Here, the elements of the partition matrix can take on values within the range $[0, 1]$. The region force term uses centroids c_k that are calculated for each cluster by finding the weighted average of the data points within that cluster. These centroids are then updated iteratively based on the current labelling. Therefore, the region force term $\|x_i - c_k\|^2$ is smaller if a node x_i is closer to the centroid c_k of a cluster and larger if it is further away. This forces the model to choose smaller ϕ_{ik} if x_i is far from cluster c_k or larger ϕ_{ik} otherwise. Thereby assigning nodes to clusters whose centroids they are sufficiently closer to.

However, in some cases, using centroids may not be very successful because the centroid of a cluster of data points may not always lie near that cluster. For example, the two moons dataset ¹ in Figure 2.1 has two clusters, red and blue, and the centroids of each lie outside their respective clusters. In the next chapter we outline a different region force term without defining centroids.

¹By courtesy of Prof. Ali Ghodsi, University of Waterloo.

Figure 2.1: Two-Moons Dataset



Chapter 3

Methodology

In this chapter we present the main models and algorithms that we use to segment cellular images. We follow the approach given in [8] which defines a new region force term to be combined with the TV and Laplacian graph partitioning problems to give two versions of the model. This region force term can be interpreted as a conditional probability for a node to belong to a certain cluster given a small set of labelled nodes. This means the models require a small set of already labelled nodes. In section 3.3 we show how to use a small set of only background pixels from the border of the cell images as the labelled nodes.

3.1 Boundary and Region Forces

Our image segmentation model can be expressed as a sum of boundary forces $B(\Phi)$ and region forces $R(\Phi)$

$$\begin{aligned} \Phi &= \arg \min_{\phi_{ik} \in [0,1]} (1 - \tau)B(\Phi) + \tau R(\Phi) \\ \text{s.t. } \Phi \mathbf{1} &= \mathbf{1}. \end{aligned} \tag{3.1}$$

where $0 \leq \tau \leq 1$ is a constant specifying the relative importance of each term. Since the elements of Φ can take in values within the range $[0, 1]$, the label for a node x_i is chosen as the cluster whose column holds the maximum value in row i of Φ . A combination of boundary and region forces are often taken because they both work in different ways to

partition a graph. Boundary forces, because they are the weights between neighbouring nodes, penalize the discontinuity between those neighbouring nodes. That is, they prefer to keep similar nodes within the same cluster and separate dissimilar nodes into different clusters. Region forces, on the other hand, penalize the inhomogeneity between clusters themselves. They can be thought of as global metrics whereas boundary forces are local metrics.

3.1.1 Boundary Force

For our boundary force term, we use the TV expression given in (2.16)

$$B(\Phi) = \sum_{k=1}^K \|\nabla \Phi_k\|. \quad (3.2)$$

The gradient operator here uses the normalized affinity matrix \hat{W} which is derived from the affinity matrix W (Section 2.3). We construct W by assigning weights between nodes using a k -Nearest Neighbour search and the Radial Basis Function defined in equation (2.1), where the distance metric used is the Euclidean distance.

3.1.2 Region Force

The region force term given in [8] is also based on the normalized affinity matrix \hat{W} and can be written as

$$R(\Phi) = \sum_{k=1}^K \sum_{x_i \in V} p_k(\hat{W}, x_i)(1 - \phi_{ik}) + (1 - p_k(\hat{W}, x_i))\phi_{ik}, \quad (3.3)$$

where $p_k(\hat{W}, x_i)$ is the conditional probability of node x_i belonging to cluster k given the labelled nodes. It is based on the idea that nodes closer to the labelled nodes should have a higher probability of having those labels. When $p_k(\hat{W}, x_i)$ is high, the first term of the region force becomes larger and in an effort to minimize this, ϕ_{ik} takes on a larger value closer to one.

To define $p_k(\hat{W}, x_i)$, suppose S_1, \dots, S_K are sets of labelled nodes in each cluster and $\hat{W}^2 = (\hat{w}_{ij}^{(2)})$ is the second power of the normalized affinity matrix given in equation (2.5).

Then $p_k(\hat{W}, x_i)$ can be defined as

$$p_k(\hat{W}, x_i) = \frac{\frac{1}{|S_k|} \sum_{j \in S_k} q_{ij}}{\sum_{r=1}^K \frac{1}{|S_r|} \sum_{j \in S_r} q_{ij}}, \quad (3.4)$$

where

$$q_{ij} = \frac{(\hat{w}_{ij}^{(2)})^2}{\hat{w}_{ii}^{(2)} \hat{w}_{jj}^{(2)}}. \quad (3.5)$$

In the case when any one of $\hat{w}_{ii}^{(2)}$ or $\hat{w}_{jj}^{(2)}$ is zero, we set $p_k(\hat{W}, x_i)$ to be equal to $1/K$. We may represent $p_k(\hat{W}, x_i)$ by an $n \times K$ matrix P such that $P_{ik} = p_k(\hat{W}, x_i)$. Also, we will denote the k -th column of P by P_k .

Combining our region and boundary forces gives our first model, the Total Variation-based model with a region force,

$$\begin{aligned} \Phi &= \arg \min_{\phi_{ik} \in [0,1]} \sum_{k=1}^K (1 - \tau) \|\nabla(\Phi_k)\| + \tau(P_k(1 - \Phi_k) + (1 - P_k)\Phi_k) \\ \text{s.t. } \Phi \mathbf{1} &= \mathbf{1}. \end{aligned} \quad (3.6)$$

Applying a quadratic relaxation on (3.6), that is, squaring the boundary term gives us

$$\begin{aligned} \Phi &= \arg \min_{\phi_{ik} \in [0,1]} \sum_{k=1}^K \frac{(1 - \tau)}{2} \|\nabla(\Phi_k)\|^2 + \tau(P_k(1 - \Phi_k) + (1 - P_k)\Phi_k) \\ \text{s.t. } \Phi \mathbf{1} &= \mathbf{1}, \end{aligned} \quad (3.7)$$

which can also be written in terms of the Laplacian (2.17) as follows

$$\begin{aligned} \Phi &= \arg \min_{\phi_{ik} \in [0,1]} \sum_{k=1}^K \frac{(1 - \tau)}{2} \Phi_k^T L \Phi_k + \tau(P_k(1 - \Phi_k) + (1 - P_k)\Phi_k) \\ \text{s.t. } \Phi \mathbf{1} &= \mathbf{1}. \end{aligned} \quad (3.8)$$

This is our second model, the Laplacian-based model with a region force. The graph Laplacian L can also be replaced with the normalized graph Laplacian L_s which is what we have used in the algorithms defined in the next section. It can be seen that the objective function of this problem is differentiable. For convenience, we divide the boundary term by 2 so that it cancels out when we have to calculate its derivative in the next section.

3.2 Algorithms for Graph Partitioning with a Region Force

In this section we outline the algorithms for solving the Laplacian-based and TV-based graph partitioning problems with the region force.

3.2.1 Laplacian-Based

The objective function of equation (3.8) is differentiable and has a convex constraint. This can be solved by a projected gradient method with Barzilai-Borwein type step sizes [12]. We denote the objective function by $J(\Phi)$. Then one iteration of the projected gradient method is given by

$$\Phi^{(j+1)} = \Pi_{\Delta}(\Phi^{(j)} - \alpha^{(j)} \partial J(\Phi^{(j)}) \Lambda^{(j)}) \quad (3.9)$$

where $\alpha^{(j)} = 1$ initially and $\Lambda^{(j)}$ is a diagonal matrix with entries on the main diagonal equal to $(\lambda_1^{(j)}, \dots, \lambda_K^{(j)})$. These are the step sizes for the gradient descent directions and they alternate between the two choices

$$\lambda_k^{(j)} = \frac{\|s_k^{(j-1)}\|^2}{\langle s_k^{(j-1)}, y_k^{(j-1)} \rangle}, \quad (3.10)$$

and

$$\lambda_k^{(j)} = \frac{\langle s_k^{(j-1)}, y_k^{(j-1)} \rangle}{\|y_k^{(j-1)}\|^2}, \quad (3.11)$$

where $s_k^{(j-1)}$ and $y_k^{(j-1)}$ are the k th columns of $s^{(j-1)} = \Phi^{(j)} - \Phi^{(j-1)}$ and $y^{(j-1)} = \partial J(\Phi^{(j)}) - \partial J(\Phi^{(j-1)})$, respectively. The operator Π_{Δ} is a projection onto the feasible domain for Φ ,

$$\Delta = \{\Phi \in \mathbb{R}^{n \times K} : \Phi \mathbf{1} = \mathbf{1}\}, \quad (3.12)$$

and it is implemented by the projection onto a simplex algorithm in [11]. This projected gradient method does not guarantee the decreasing of the objective function. So to ensure sufficient decreasing, a non-monotone line search is taken by decreasing the parameter $\alpha^{(j)}$ in equation (3.9), based on an Armijo-type acceptability test [12]

$$J(\Phi^{(j+1)}) \leq J(\Phi^{(j)}) + \theta Tr(\partial J(\Phi^{(j)})^T s^{(j)}), \quad (3.13)$$

for some small constant $\theta > 0$. The stopping criterion is chosen such that two consecutive objective function values are close enough

$$|J(\Phi^{(j+1)}) - J(\Phi^{(j)})| \leq \epsilon J(\Phi^{(j)}), \quad (3.14)$$

where $\epsilon > 0$ is a small constant. A summary of the algorithm, called LapRF, is provided in Algorithm 1. It is to be noted that we use the normalized graph Laplacian L_s in the computation.

Algorithm 1 Laplacian-based multi-class graph partitioning with a region force (LapRF)

Require: $L_s, P = (p_{ik}), \tau$

Ensure: Φ

```

1: procedure LAPRF
2:   while (3.14) is not satisfied do
3:      $\alpha = 1$ ;
4:     calculate  $\Phi$  by (3.9)
5:     while (3.13) is not satisfied do
6:        $\alpha = 0.8\alpha$ ;
7:       recalculate  $\Phi$  by (3.9)
8:     end while
9:   end while
10: return  $\Phi$ 
11: end procedure

```

3.2.2 TV-Based

Next we outline the algorithm for solving equation (3.6). By noting the variational formulation for total variation

$$\|\Phi_k\|_1 = \max_{q_k \in \mathbb{R}^{n \times n}, \|q_k\|_\infty \leq 1} \langle \Phi_k, \text{div} q_k \rangle, \quad (3.15)$$

the minimization problem can be expressed as a saddle-point problem,

$$\min_{\Phi \in \Delta} \max_{\|q_k\|_\infty \leq 1} \sum_{k=1}^K (1 - \tau) \langle \Phi_k, \text{div} q_k \rangle + \tau (P_k(1 - \Phi_k) + (1 - P_k)\Phi_k). \quad (3.16)$$

It can be solved by a primal-dual hybrid gradient method which alternates between gradient ascent for the dual variable q_k and gradient descent for the primal variable Φ_k [13]. The steps are defined as

$$q_k^{(j)} = \Pi_{\|q_k\|_\infty \leq 1} (q_k^{(j-1)} - \beta^{(j-1)} \nabla \Phi_k^{(j-1)}) \text{ for } k = 1, \dots, K, \quad (3.17)$$

$$\Phi^{(j)} = \Pi_{\Delta}(\Phi^{(j-1)} - \gamma^{(j-1)}((1 - \tau)divQ^{(j)} + \tau(1 - 2P))), \quad (3.18)$$

where β and γ are step sizes following the theoretical analysis in [14] and $Q = [q_1, \dots, q_K]$. The projection $\Pi_{\|q_k\|_{\infty} \leq 1}$ onto $\|q_k\|_{\infty} \leq 1$ is done by hard thresholding and Π_{Δ} is the projection onto a simplex given in [11]. The stopping criterion is the same as (3.14). A summary of the algorithm, TVRF, is given in Algorithm 2.

Algorithm 2 TV-based multi-class graph partitioning with a region force (TVRF)

Require: $P = (p_{ik}), W, \{\beta_l = 0.2l\}, \{\gamma_l = \frac{0.1l}{(1+0.1l)}\}, \tau$

Ensure: Φ

```

1: procedure TVRF
2:    $l = 0$ 
3:   while (3.14) is not satisfied do
4:     for  $k=1, \dots, K$  do
5:       calculate  $\nabla\Phi_k$ ,
6:        $q_k = \Pi_{\|q_k\|_{\infty} \leq 1}(q_k - \beta_l \nabla\Phi_k)$ ,
7:     end for
8:     calculate  $divQ$ ,
9:      $\Phi = \Pi_{\Delta}(\Phi - \gamma_l((1 - \tau)divQ + \tau(1 - 2P)))$ ,
10:     $l = l + 1$ ,
11:  end while
12: return  $\Phi$ 
13: end procedure

```

3.3 Cellular Image Considerations

As discussed earlier, the probability matrix P requires a set of labelled nodes from all clusters. In the case of cell images, it is usually not practical to have a set of labelled cell and background pixels to start off with. However, it is possible to have only the background on a thin border around the image. This can be done taking the image of a sample of cells in such a way that the cells are positioned within a thin border of only the background substance they are suspended in. This border can be taken as our labelled set of pixels S_B . Since the labelled set is only on the border of the image, we choose to first calculate W by only taking the pixel intensities in the data vectors, not locations. We then calculate $\sum_{j \in S_B} q_{ij}$ for each node. If it is zero, we label the node as cell and if it's not, we label it as background. This gives us an “extreme” labelling in which only those pixels that are very

different in intensity from the labelled background pixels are labelled as cell. Now that we have a set of labelled background and cell pixels, we use half of these to calculate P , this time with W constructed by taking both pixel intensities and locations. We use half because experimentally we found it produced best result. Finally, this P and W is used for graph-partitioning.

It can be seen from (3.9) and (3.11) that for initialization of Φ , LapRF requires two initial partition matrices. We choose Φ^0 to be all background nodes and Φ^1 to be all cell nodes. TVRF requires a single initial partition matrix ((3.17), (3.18)) which we chose to be all background nodes.

Chapter 4

Numerical Results

In this chapter, we illustrate the numerical results of the algorithms discussed in Chapter 3. We apply those algorithms to various cell images including bright-field images and 2- and 3-D fluorescent cell images. Labelled pixels were obtained from the border of the images and in cases when the borders contained cell pixels, they were manually removed.

While there are a number of parameters, not all of them were required to change for different types of images. We fixed $\epsilon = 10^{-6}$ (3.14) for the stopping criteria and $\theta = 10^{-6}$ (3.13). The remaining parameters ω (2.1), τ (3.1), and the number of neighbours k used to construct W (Section 2.2), will be given with the results. After obtaining the initial segmentation results, we post-process them by bridging unconnected cell pixels [15], filling in holes within the cells and removing small cell fragments from the background. The results were obtained using MATLAB on a MAC machine with a 2.5GHz processor.

4.1 Bright-Field Images

The bright-field images are 512×512 pixel frames of C2C12 (muscle) cells obtained from the Department of Medicine and Human Genetics at McGill University. Our parameters here are $\omega = 50$, $k = 40$, $\tau = 0.1$ for LapRF, and $\tau = 0.3$ for TVRF. Figure 4.1 shows the original bright-field image and Figure 4.2 shows the segmentation results using our Laplacian and TV-based algorithms with Region Force. The first row of Figure 4.2 shows the raw results before any post-processing is applied. The second row shows the final results after post-processing and the third row is a visualization of the final segmentation result on the original image. The segmentation results given by the Laplacian and TV models are

shown on the left and right columns, respectively. We can see from the raw segmentation results that the TV model captures more of the cells as compared to the Laplacian model. However, it also under segments in some areas, for instance, the left leg of the bottom left cell. The Laplacian-based model, on the other hand, captures more of the fine details in the shape of the cell, as can be seen in the top right cell. It produces results with more “holes” but they can be easily filled in by a simple post-processing step. Therefore, in the final segmentation results, Laplacian slightly over-segments and TV slightly under-segments but both results are very much comparable.

Their running times, however, differ considerably and are shown in Table 4.1. When applied on our bright-field image, LapRF took 23.8 seconds and TVRF 25110.3 seconds, about a thousand times longer than LapRF. This is due to the computational complexity of calculating the gradient and divergence in (3.17) and (3.18), respectively.

	Time
Constructing W and P	118.2 sec
LapRF	23.8 sec
TVRF	25110.3 sec

Table 4.1: The time comparison of our algorithms tested on a bright-field image.

4.2 ISBI 2013 Cell Tracking Challenge Datasets

The IEEE International Symposium on Biomedical Imaging 2013 Cell Tracking Challenge [16] was established to evaluate different automated cell tracking and segmentation algorithms. In this section, we apply our algorithm on their datasets and compare our results with the other algorithms given in [16]. The evaluation metric used for segmentation is the Jaccard Similarity Index given by

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (4.1)$$

where A is the set of pixels in the ground truth and B is the set of pixels in the segmented result. We can see that if the segmentation matches the ground truth exactly then $|A \cap B| = |A \cup B|$ and $J(A, B) = 1$. Similarly, if there is no overlap between them, $|A \cap B| = 0$, and $J(A, B) = 0$.

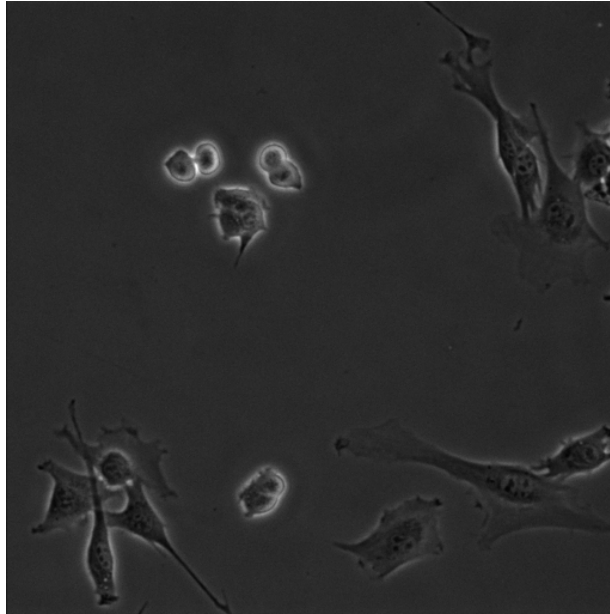


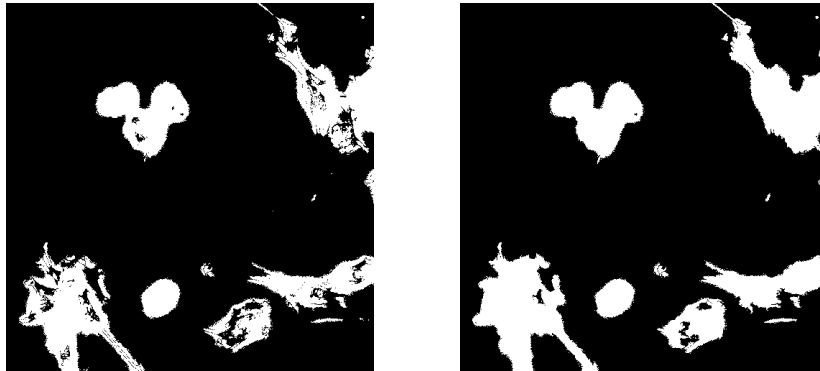
Figure 4.1: A bright-field image of C2C12 cells.

The datasets are cell videos made up of a number of still images. Due to faster performance, we tested only the Laplacian-based algorithm on six videos of the N2DH-SIM simulated fluorescent images. Table 4.2 shows the comparison of LapRF with the algorithms/teams of the ISBI 2013 Cell Tracking Challenge. It can be seen that LapRF performs better than or as well as others on videos 02, 04 and 05 and has comparable results on the other videos as well. Sample images and our segmentation results of each video are also shown in Figures 4.3 to 4.8.

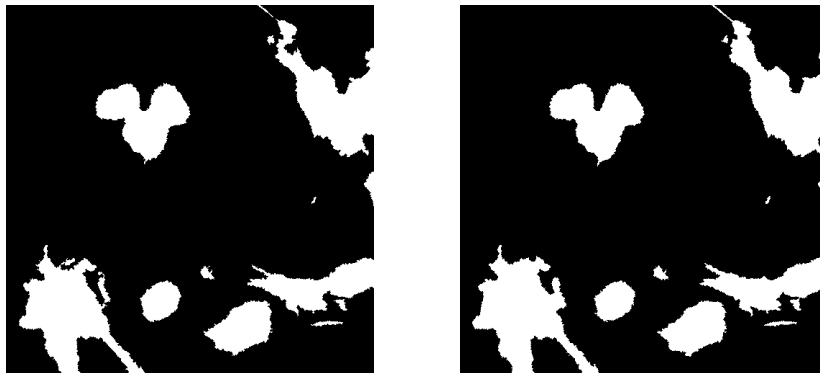
4.3 Cell Tracking Challenge 2017 Datasets

Finally, we test our Laplacian-based algorithm on two and three-dimensional images from the 2017 Cell Tracking Challenge¹. We perform tests on various cell datasets with different properties to evaluate the strengths and weaknesses of our algorithm.

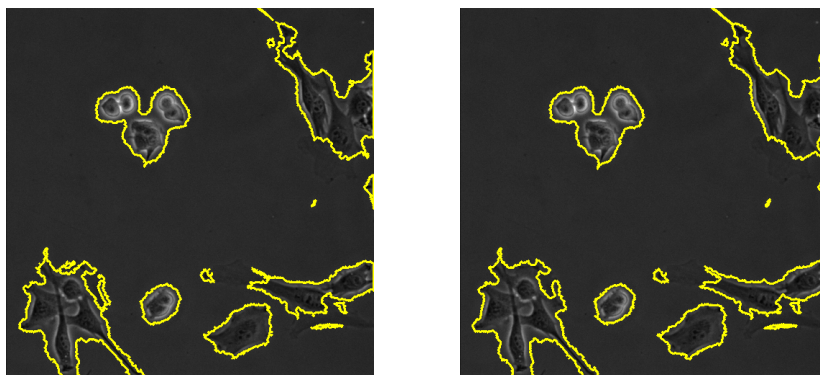
¹Challenge Website: <http://www.codesolorzano.com/Challenges/CTC/Datasets.html>



(a) Raw segmentation results



(b) Final Segmentation after post-processing



(c) Visualization

Figure 4.2: Bright-field image segmentation given by (left column) Laplacian-based model and (right column) TV-based model.

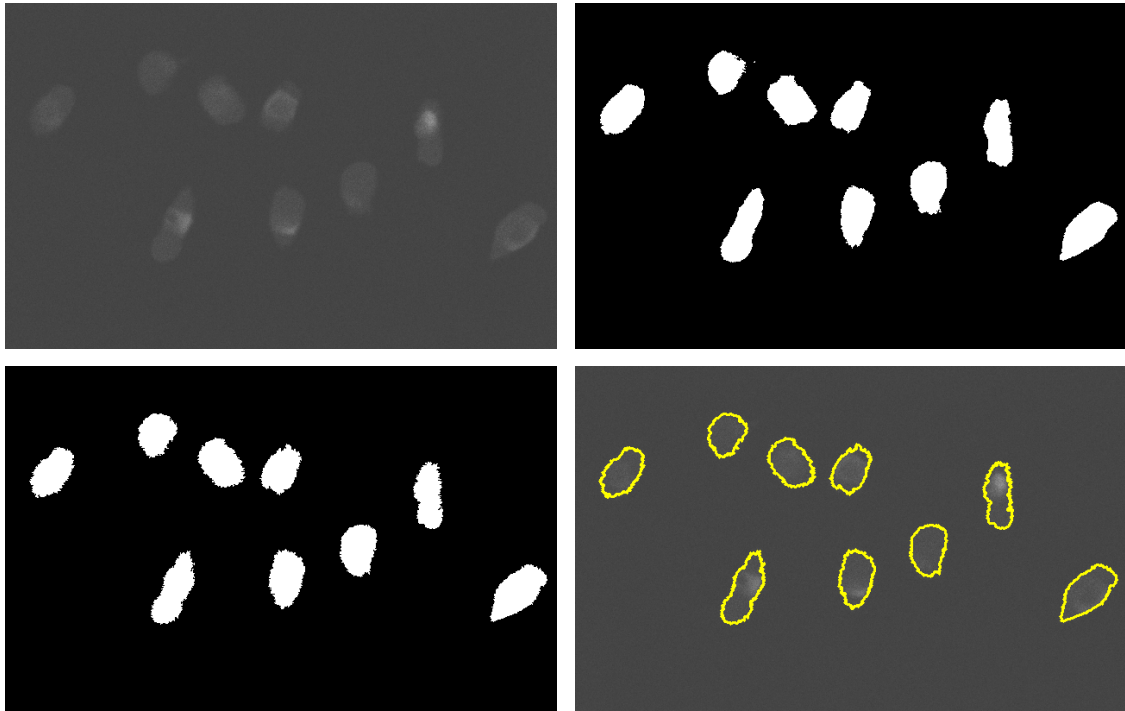


Figure 4.3: Segmentation on an image from N2DH-SIM dataset – video 01. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 40$, $\tau = 0.1$.

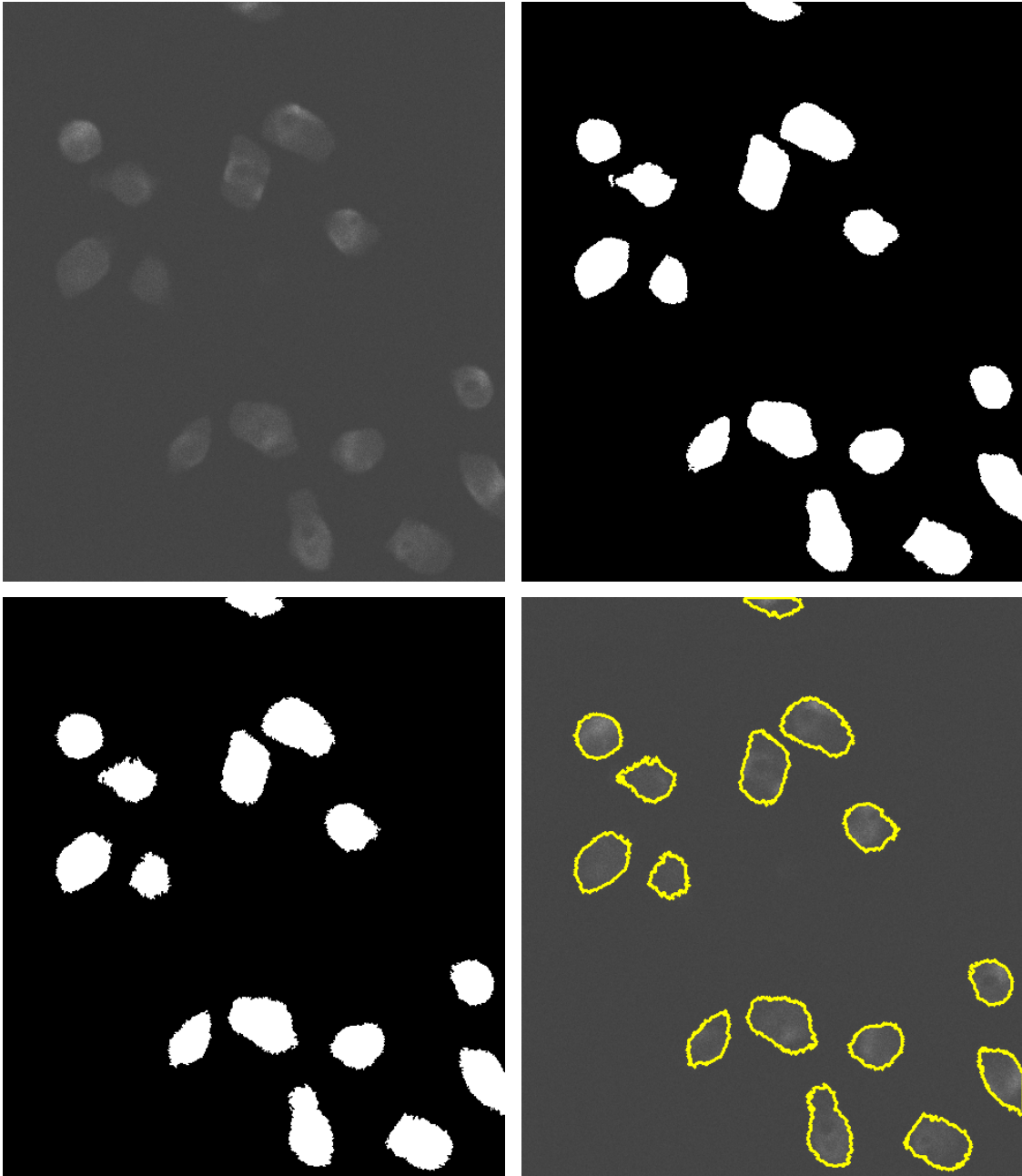


Figure 4.4: Segmentation on an image from N2DH-SIM dataset – video 02. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 40$, $\tau = 0.1$.

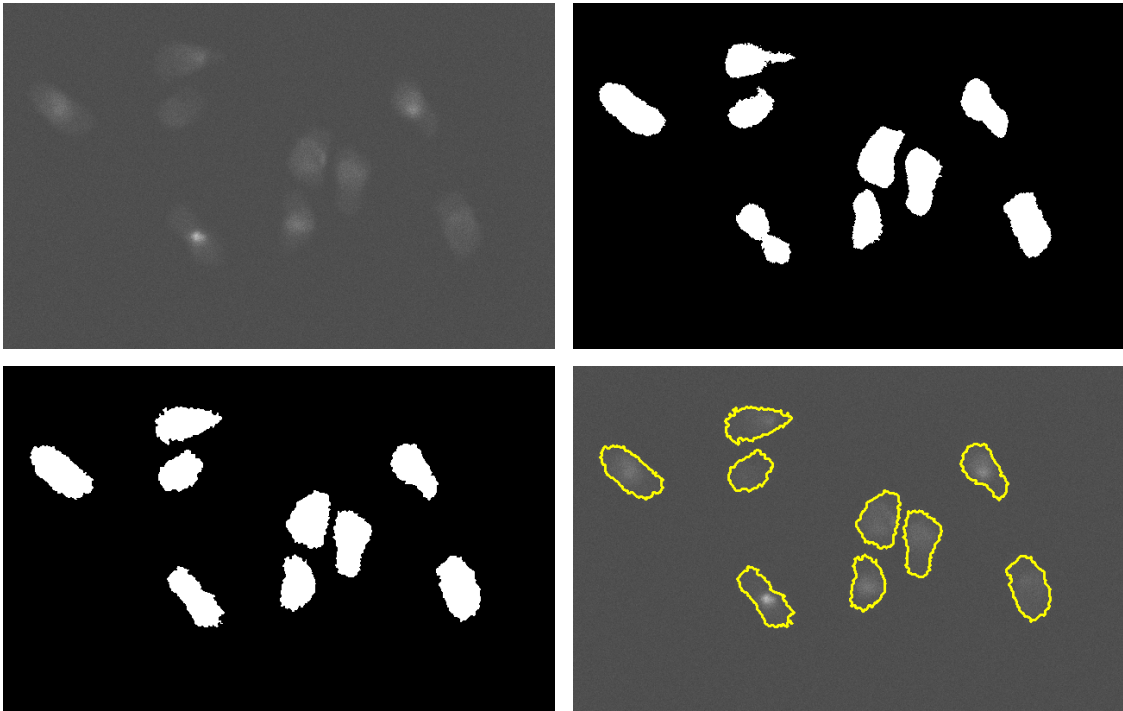


Figure 4.5: Segmentation on an image from N2DH-SIM dataset – video 03. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 40$, $\tau = 0.1$.

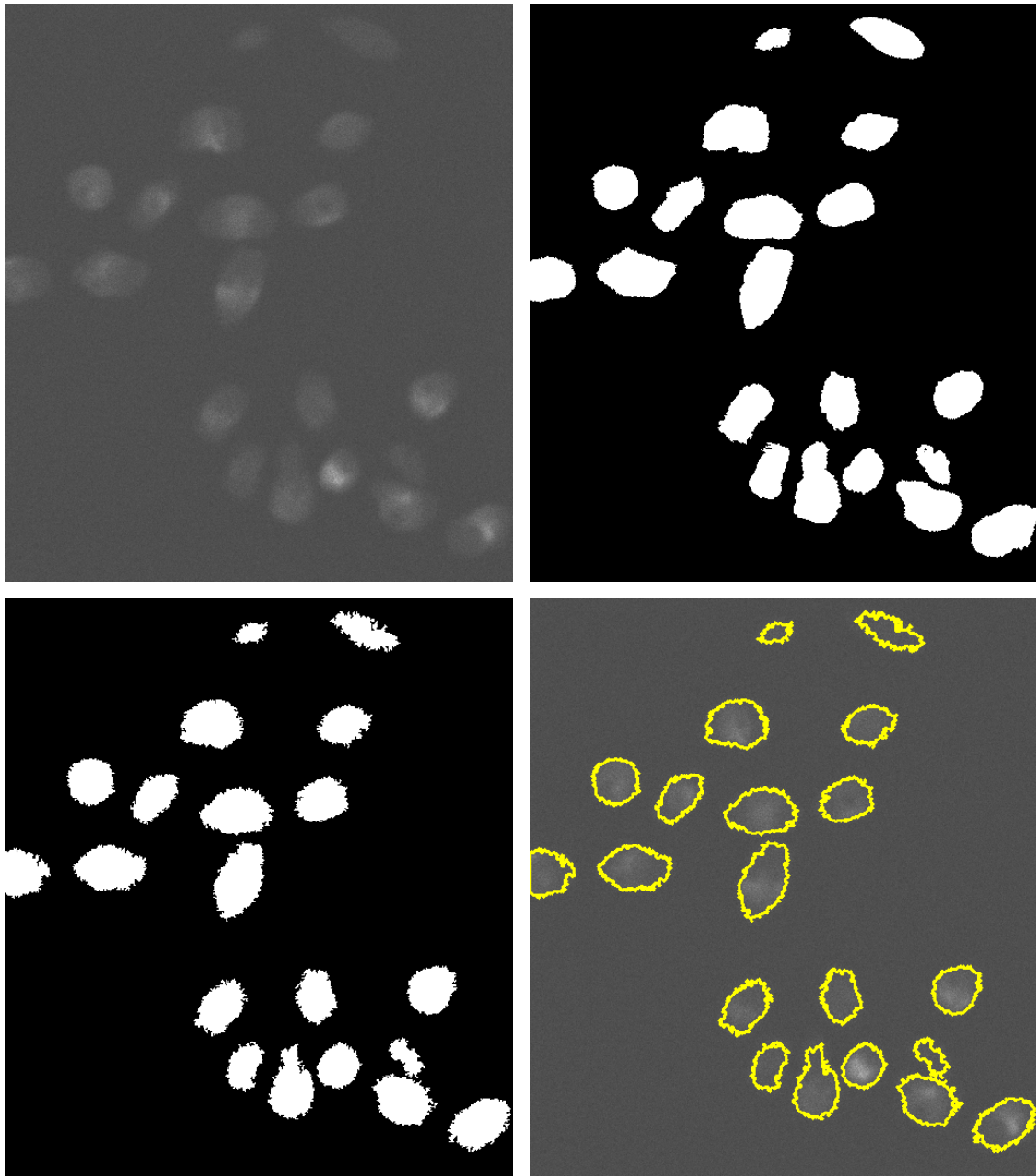


Figure 4.6: Segmentation on an image from N2DH-SIM dataset – video 04. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 40$, $\tau = 0.3$.

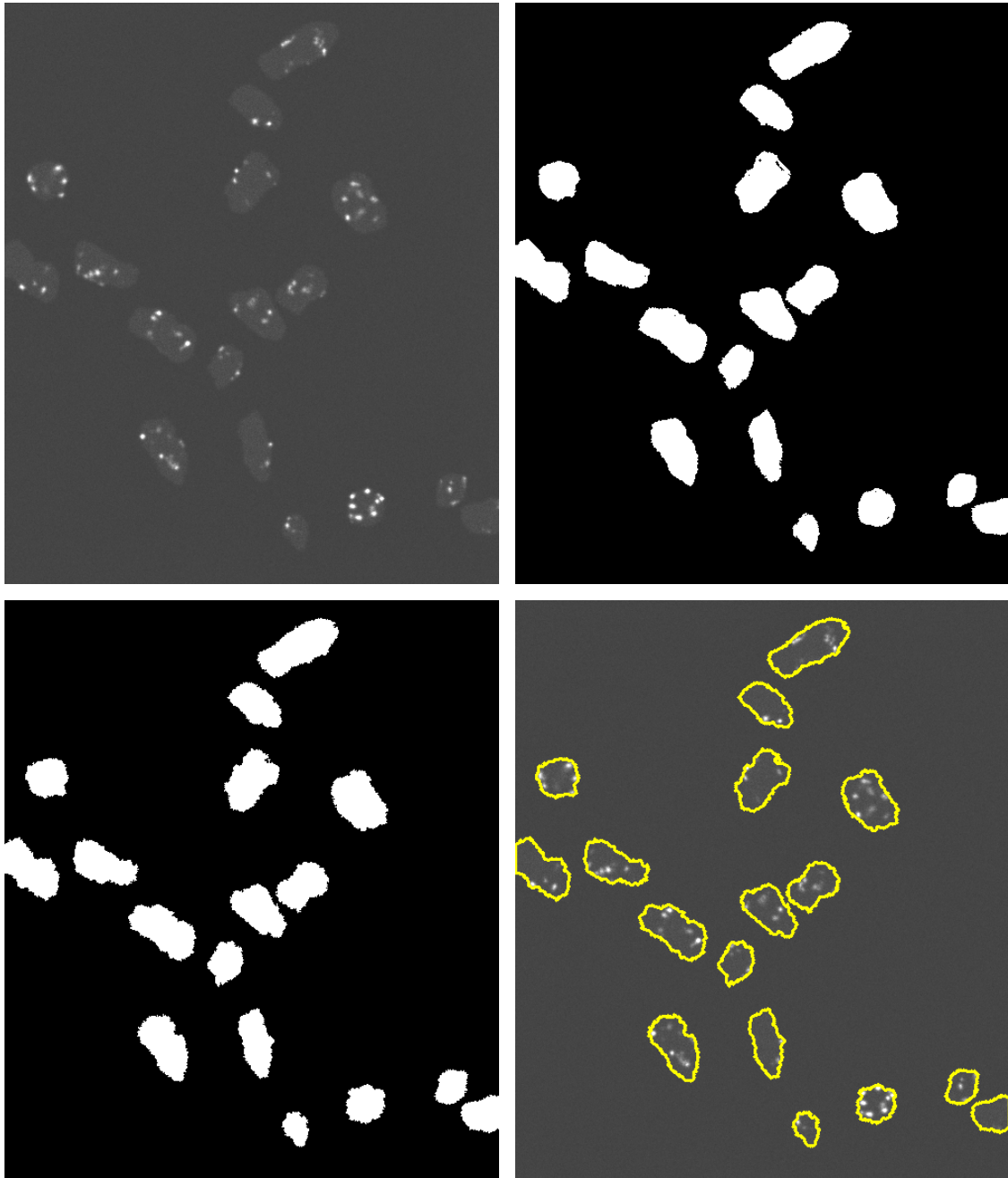


Figure 4.7: Segmentation on an image from N2DH-SIM dataset, – video 05. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 40$, $\tau = 0.1$.

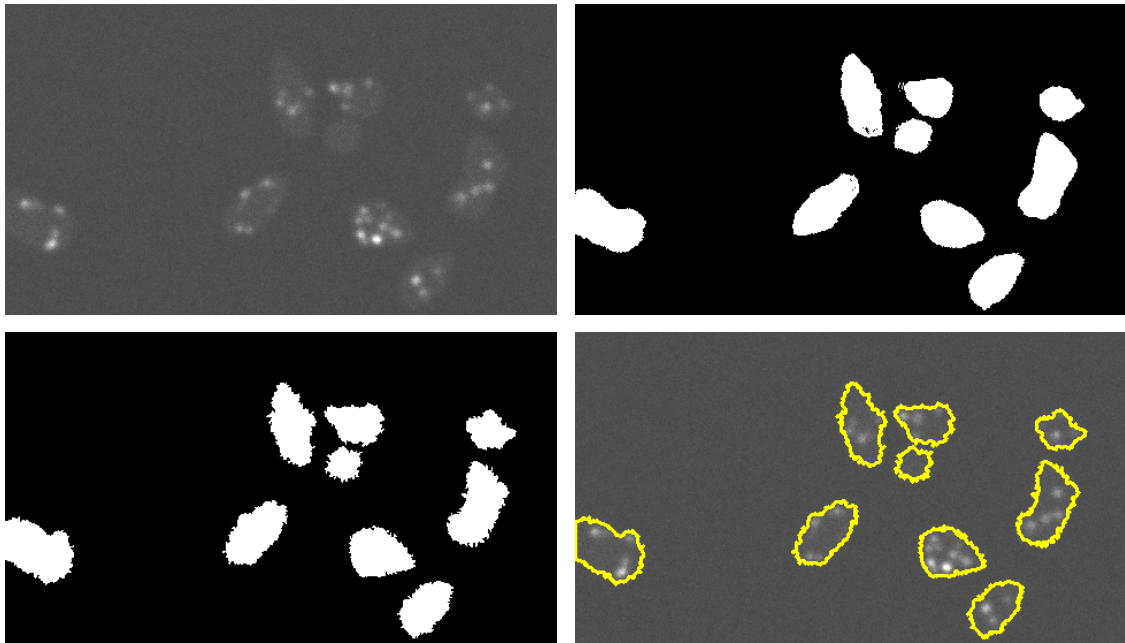


Figure 4.8: Segmentation on an image from N2DH-SIM dataset – video 06. Top row: original image, ground truth. Bottom Row: final segmentation, visualization. $\omega = 10$, $k = 20$, $\tau = 0.3$.

Video	01	02	03	04	05	06
COM-US	0.75	0.77	0.65	0.64	0.64	0.75
HEID-GE	0.91	0.9	0.82	0.83	0.81	0.89
KTH-SE	0.91	0.89	0.84	0.83	0.8	0.83
LEID-NL	0.93	0.91	0.83	0.84	0.84	0.91
PRAG-CZ	0.8	0.8	0.75	0.78	0.76	0.79
UPM-ES	0.91	0.89	0.81	0.82	0.77	0.86
LapRF	0.90	0.91	0.83	0.86	0.89	0.88

Table 4.2: A comparison of the segmentation scores of ISBI 2013 Cell Tracking Challenge algorithms and LapRF. Bolded entries are the highest scores for that video.

4.3.1 2-D Images

The first image we test our algorithm on is a 839×992 pixel image from the Fluo-C2DL-MSD dataset. These are fluorescent images of rat mesenchymal stem cells. The original image and our segmentation results are given in Figure 4.9. It can be seen that the algorithm is able to effectively segment cells that are irregularly shaped, as opposed to most other images which contain more rounded, convex shaped cells. Another fluorescent image is from the Fluo-N2DH-GOWT1 dataset which is a 1024×1024 image of mouse stem cells. The segmentation results are given in Figure 4.10. The cells here are much more uniform but vary greatly in intensity. For example, the pair of joined cells in the bottom right are almost too dark to be noticeable from the background, but the algorithm successfully captures them along with the brighter cells. The last fluorescent image is a 700×1100 pixel image from the Fluo-N2DL-HeLa dataset and the segmentation results can be seen in Figure 4.11. These HeLa cells are very densely packed and our algorithm struggles to correctly identify individual cells in some areas.

Our next two datasets are obtained by Phase-Contrast microscopy. The first are 520×696 images of Glioblastoma-astrocytoma U373 cells from the PhC-C2DH-U373 dataset. The results are shown in Figure 4.12. While the algorithm is able to identify the general location of each cell, the inconsistent halos surrounding the cells makes it difficult to accurately determine the boundaries of cells. Our final 2-D image is from the PhC-C2DL-PSC dataset which contains 576×720 pixel images of pancreatic stem cells. The image and results are given in Figure 4.13. It can be seen that the background of this cell image is very inconsistent and merely taking the border as the labelled background pixels does not work very well in this case. Therefore, to obtain a reasonable result, we had to take an additional

small sample of background pixels from the centre of the image.

4.3.2 3-D Images

The three-dimensional images are from the FLUO-N3DH-SIM+ simulated dataset which consist of 59 slices, where each slice is a 350×640 two-dimensional image. Figure 4.14 shows some of the slices of one 3-D image. In order to work on 3-D images, our algorithm is modified to include the 3-D location of pixels in the data vectors when calculating the affinity matrix W . We compare the result of this 3-D segmentation with the result of performing 2-D segmentation on each slice. For computation within a reasonable amount of time, we have resized each slice to 233×426 pixel. Figure 4.15 shows the results which are very much comparable. However, upon closer inspection of a single slice in Figure 4.16, it can be seen that 3-D segmentation is able to capture more cell pixels that are of especially low contrast. This could be due to extra information from neighbouring slices. The running time for 3-D segmentation is also less than that of 2-D segmentation, given in Table 4.3.

	Time
2-D segmentation on each slice	1859.8 sec
3-D segmentation on whole image	1017.3 sec

Table 4.3: A time comparison of 2- and 3-D segmentation on 3-D image.

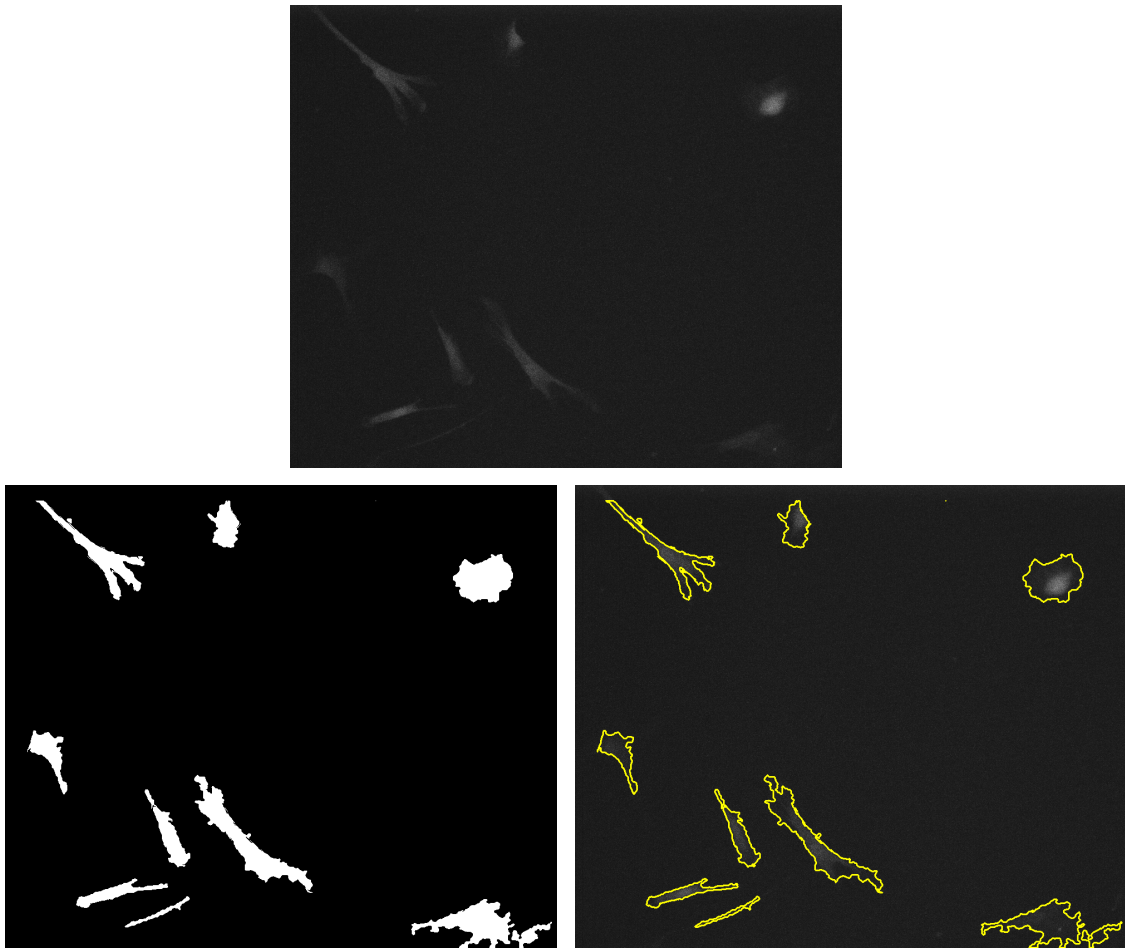


Figure 4.9: Top: Fluo-C2DL-MSc original image. Bottom: (left) final segmentation result, (right) visualization. $\omega = 10$, $k = 40$, $\tau = 0.1$.

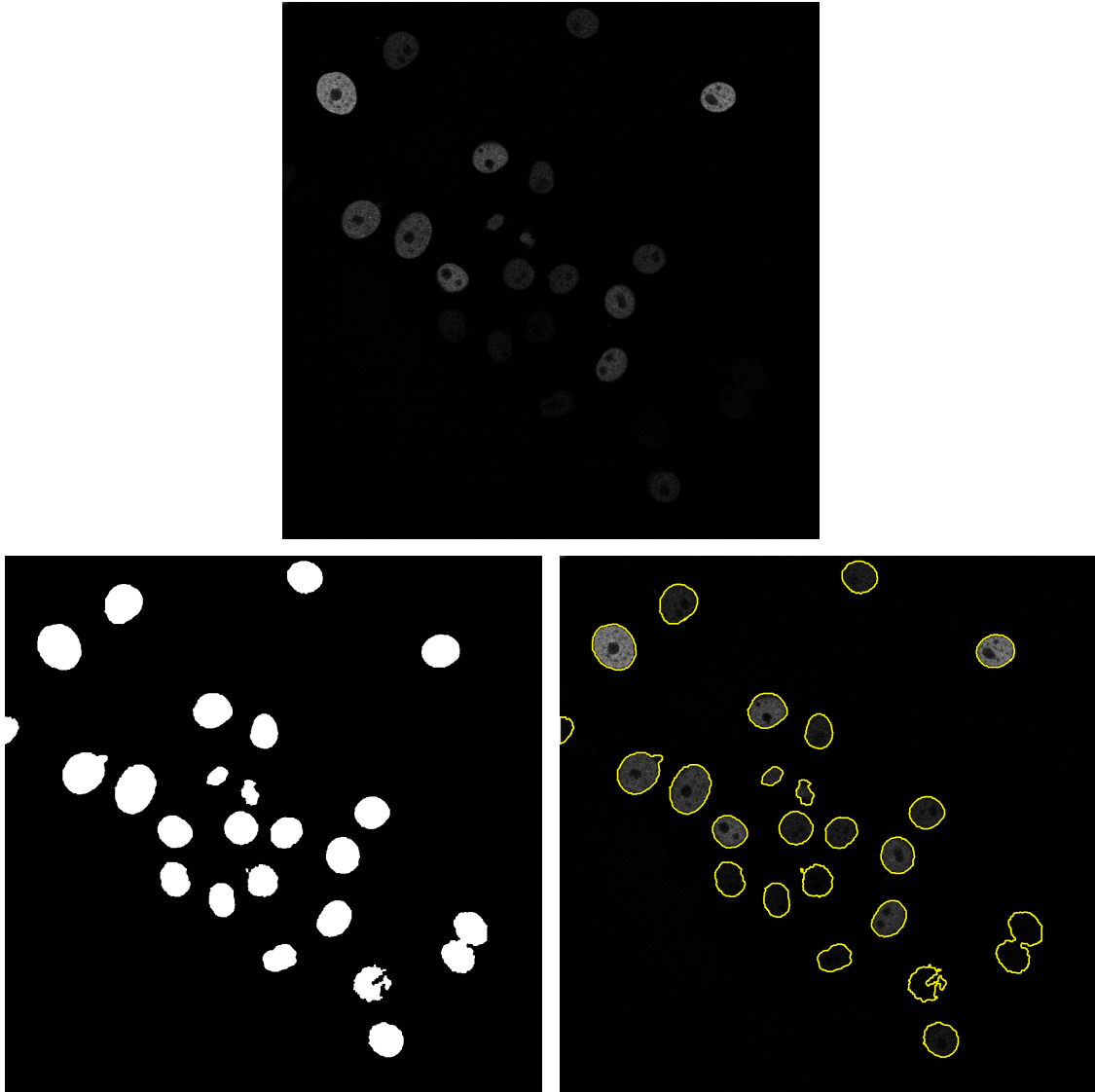


Figure 4.10: Top: Fluo-N2DH-GOWT1 original image. Bottom: (left) final segmentation result, (right) visualization. $\omega = 1$, $k = 40$, $\tau = 0.005$.

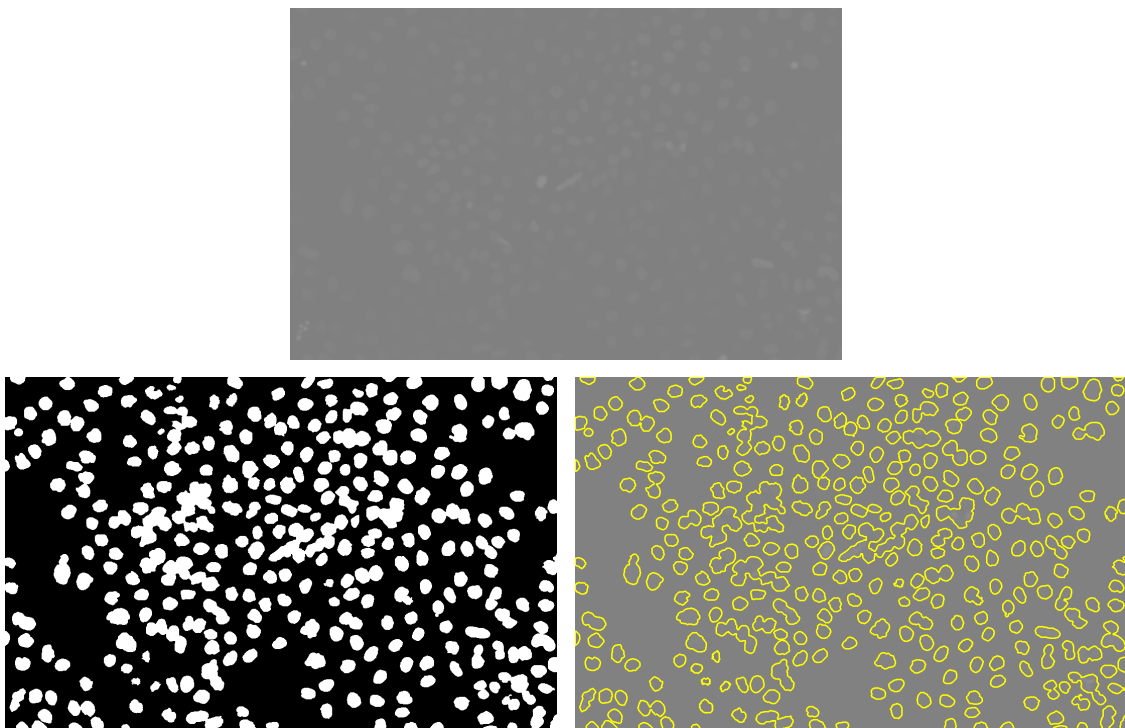


Figure 4.11: Top: Fluo-N2DL-HeLa original image. Bottom: (left) final segmentation result, (right) visualization. $\omega = 10$, $k = 40$, $\tau = 0.005$.

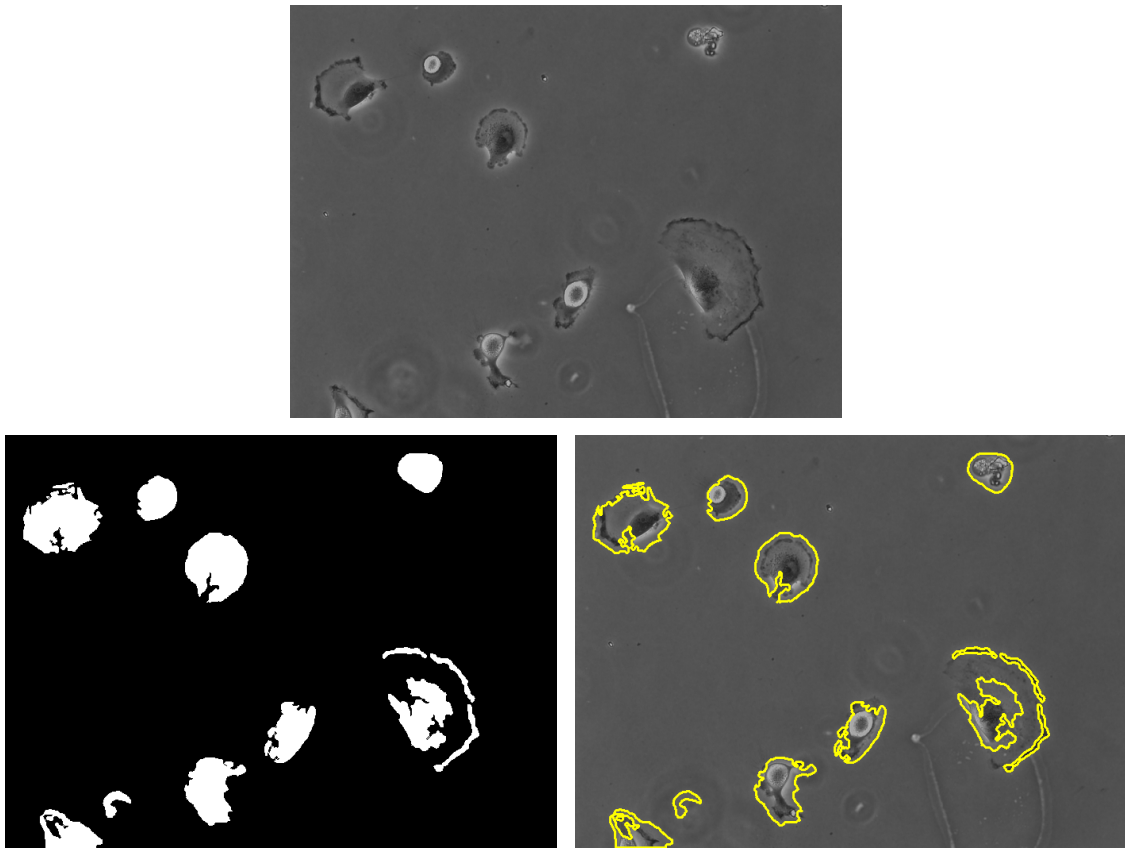


Figure 4.12: Top: PhC-C2DH-U373 original image. Bottom: (left) final segmentation result, (right) visualization. $\omega = 10$, $k = 40$, $\tau = 0.005$.

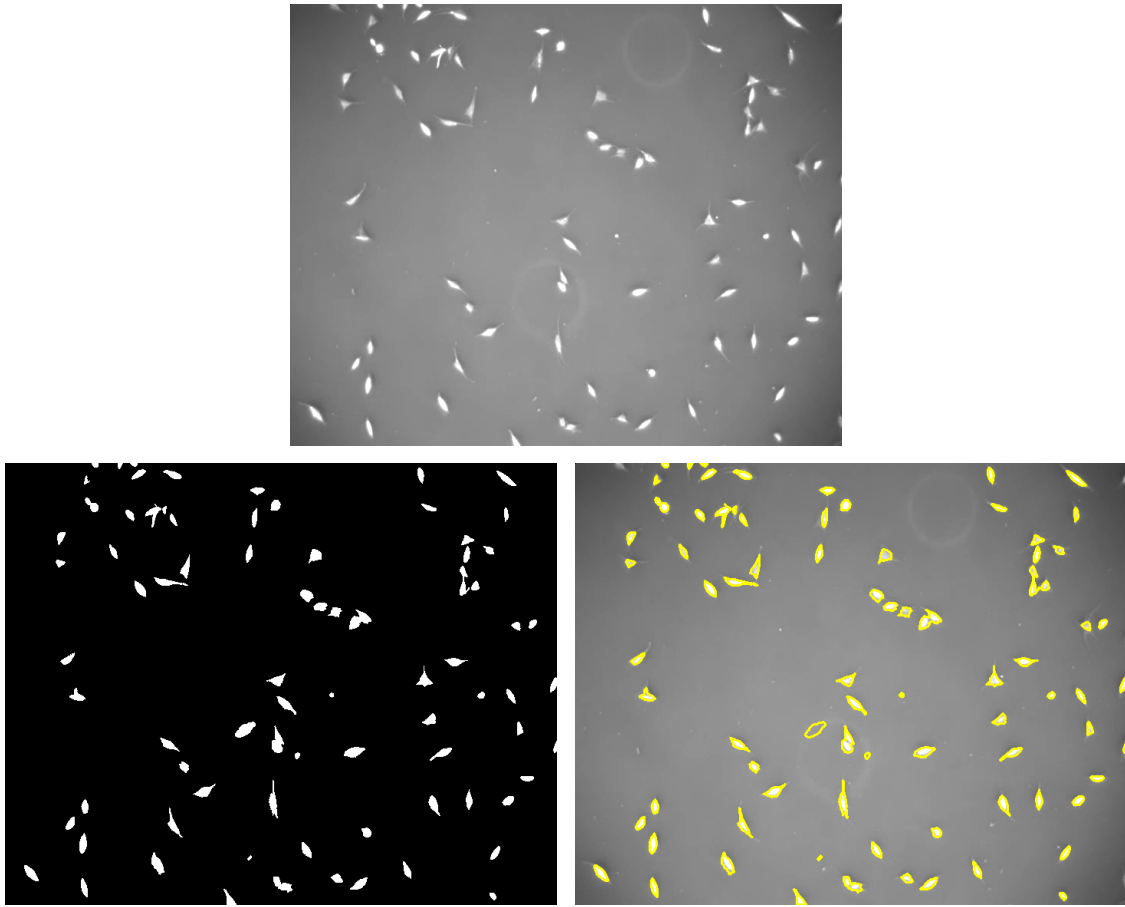


Figure 4.13: Top: PhC-C2DL-PSC original image. Bottom: (left) final segmentation result, (right) visualization. $\omega = 10$, $k = 25$, $\tau = 0.005$.

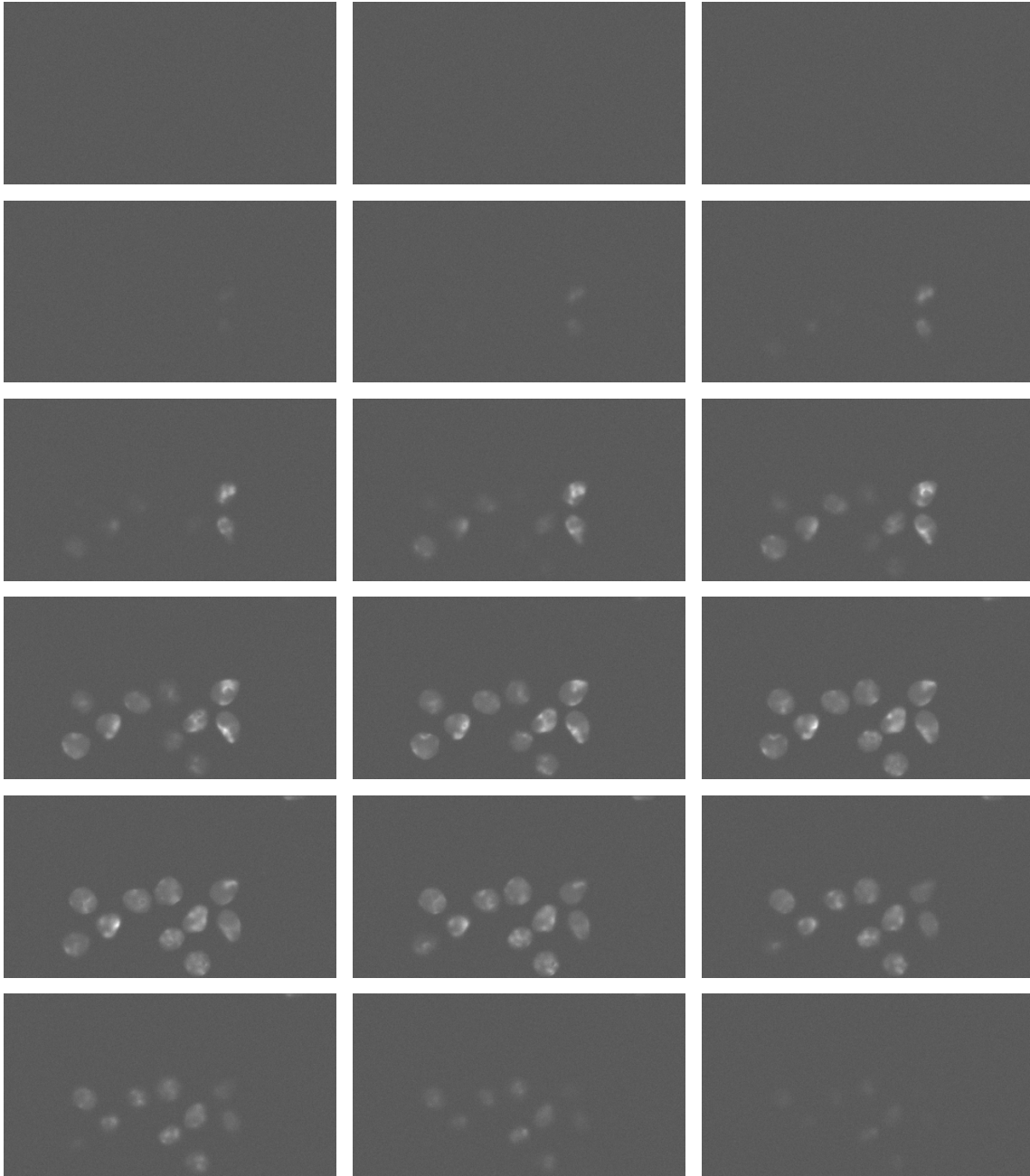
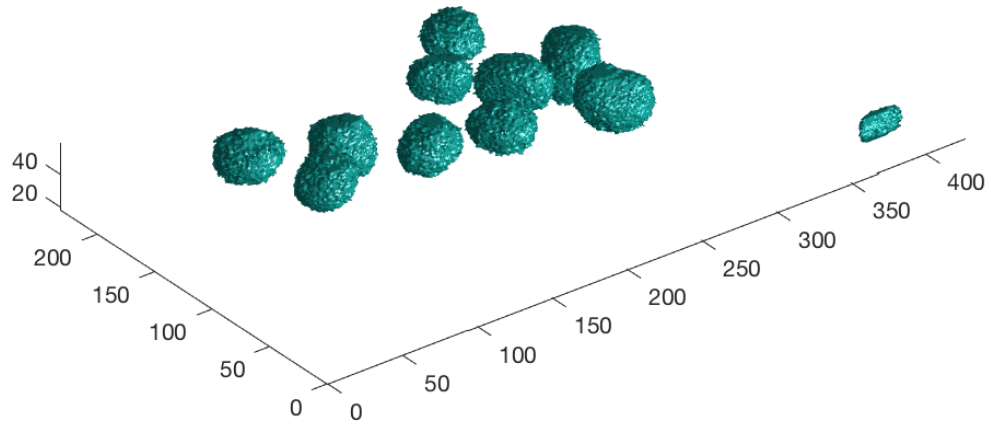
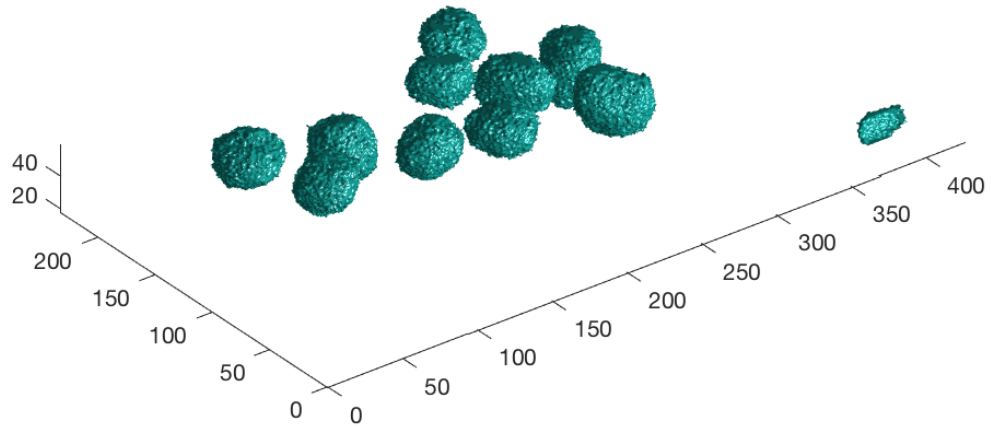


Figure 4.14: From top left to right then next rows: slices number 8, 11, 14, ..., 59 of a FLUO-N3DH-SIM+ 3-D image.



(a) 2-D Segmentation on each slice. $\omega = 1, k = 70, \tau = 0.1$.



(b) 3-D Segmentation on whole image. $\omega = 10, k = 10, \tau = 0.1$.

Figure 4.15: Results of 2- and 3-D segmentation applied on a FLUO-N3DH-SIM+ 3-D image.

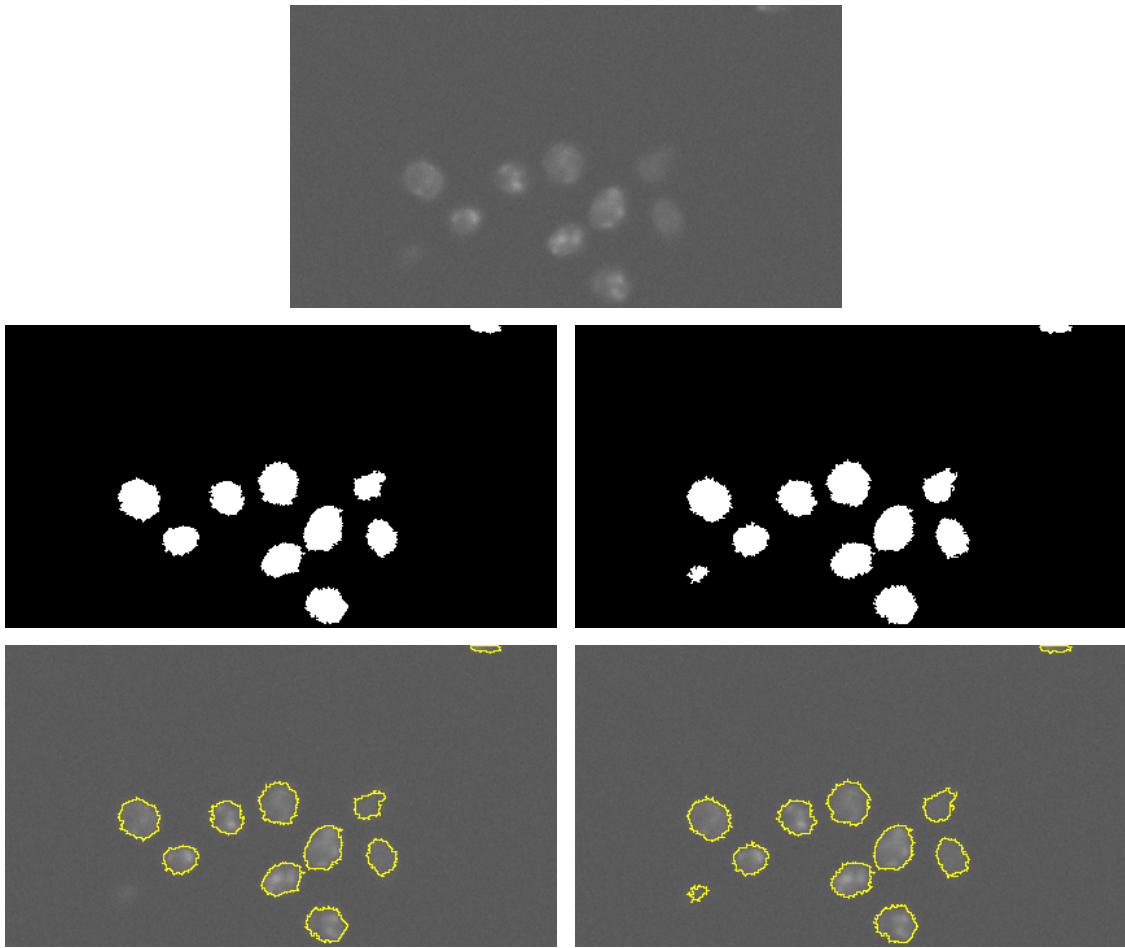


Figure 4.16: Top: Slice number 52 of 3-D image. Second row: 2-D segmentation result, 3-D segmentation result. Third row: 2-D segmentation visualization, 3-D segmentation visualization.

Chapter 5

Conclusion

In this work we presented two algorithms for segmentation of cellular images using a graph-partitioning approach. The first, a Laplacian based model, was shown to give good results on the challenging bright-field images as well as other datasets within a reasonable amount of time. The second model, based on the Total Variation, was shown to give equally good results on bright-field images but was very computationally expensive. Furthermore, both models were capable of successfully segmenting images containing multiple cells with minimal parameter tuning.

The Laplacian-based algorithm was also tested on six datasets from the ISBI 2013 Cell Tracking Challenge and six datasets from the Cell Tracking Challenge 2017. For the first six datasets, we evaluated the segmentation accuracy using the Jaccard Similarity Index. Our algorithm was seen to perform the best on three of the datasets and produced comparable results on the other three as well. For the last six datasets, we carefully picked unique types of cells and cell images to determine the strengths and weaknesses of our algorithm. Our results on the Fluo-C2DL-MSD and Fluo-N2DH-GOWT1 datasets demonstrate that the algorithm is able to effectively segment irregularly shaped cells as well as extremely low intensity cells. Furthermore, we showed that the algorithm is faster and better able to segment 3-D images as a whole than segmenting each slice separately.

A couple of limitations of the algorithm were discovered by testing on the Fluo-N2DL-HeLa dataset where cells were very densely packed together and resulted in joined cells, and the PhC-C2DL-PSC dataset where the background is very inconsistent. Further work can be done in the investigation of under-segmentation on densely packed cell images as well as methods to extract an accurate background labelling when backgrounds are incon-

sistent. Another area for investigation could be in reducing the computational complexity of the Total Variation based model so that it can be tested on more data sets in less time.

References

- [1] Bernhard Schörlkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.
- [2] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601-1608, 2004.
- [3] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35-43, 2001.
- [4] Fan Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, December 1996.
- [5] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005-1028, 2008.
- [6] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266-277, February 2001.
- [7] Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi. Multi-class graph Mumford-Shah model for plume detection using the MBO scheme. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 209-222. Springer, 2015.
- [8] Ke Yin, Xue-Cheng Tai and Stanley Osher. An Effective Region Force for Some Variational Models for Learning and Clustering. In *Technical Report, CAM report 16-18, UCLA*, 2016.
- [9] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888-905, August 2000

- [10] O. L’ezoray, A. Elmoataz, and V. T. Ta. Nonlocal PDEs on graphs for active contours models with applications to image segmentation and data clustering. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 873-876, March 2012.
- [11] Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.
- [12] Yu-Hong Dai and Roger Fletcher. Projected Barzilai-Borwein methods for largescale box-constrained quadratic programming. *Numerische Mathematik*, 100(1):21- 47, March 2005.
- [13] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, pages 08-34, 2008.
- [14] Silvia Bonettini and Valeria Ruggiero. On the Convergence of Primal Dual Hybrid Gradient Algorithms for Total Variation Image Restoration. *Journal of Mathematical Imaging and Vision*, 44(3):236-253, January 2012.
- [15] MathWorks®. Image Processing Toolbox™: Reference (R2017a). Retrieved July 19, 2017 from https://www.mathworks.com/help/pdf_doc/images/images_ref.pdf, page 209.
- [16] M. Maska, V. Ulman, D. Svoboda, P. Matula, P. Matula, C. Ederra, A. Urbiola, T. Espana, S. Venkatesan, D. M. W. Balak, P. Karas, T. Bolckova, M. Streitova, C. Carthel, S. Coraluppi, N. Harder, K. Rohr, K. E. G. Magnusson, J. Jalden, H. M. Blau, O. Dzyubachyk, P. Krizek, G. M. Hagen, D. Pastor-Escuredo, D. Jimenez-Carretero, M. J. Ledesma-Carbayo, A. Munoz-Barrutia, E. Meijering, M. Kozubek, and C. Ortiz-de Solorzano. A benchmark for comparison of cell tracking algorithms, *Bioinformatics*, 30 (2014), pp. 1609-1617.
- [17] E. Meijering, O. Dzyubachyk, I. Smal, W. A. van Cappellen. Tracking in Cell and Developmental Biology. *Seminars in Cell and Developmental Biology*, vol. 20, no. 8, October 2009, pp. 894-902.
- [18] 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.
- [19] L. Bradbury. Segmentation of Bright-field Cell Images. Master’s thesis, University of Waterloo, 2009.

- [20] Marko Tscherepanow, Frank Zöllner, and Franz Kummert. Automatic Segmentation of Unstained Living Cells in Bright-Field Microscope Images. *Proceedings of the Workshop on Mass-Data Analysis of Images and Signals (MDA2006)*, pp. 86-95, 2006.
- [21] Christian Maurer, Alexander Jesacher, Stefan Bernet and Monika Ritsch-Marte. Phase contrast microscopy with full numerical aperture illumination. *Opt. Expr.* *16 (2008)*, pp. 19821–19829.
- [22] Soo Min Kang. Automated Segmentation of Brightfield Images using the Bhattacharyya Measure via Graph-Cut Segmentation. Master’s thesis, University of Waterloo, 2012.
- [23] Ian B. Dimock. Cellular Image Segmentation using N-agent Cooperative Game Theory. Master’s thesis, University of Waterloo, 2015.