UNIVERSITY OF CALIFORNIA, SAN DIEGO

Automated Cell Detection and Morphology Analysis on Microscopic Images in Imaging Flow Cytometry

A thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Electrical Engineering (Signal and Image Processing)

by

Nirja Jatin Mehta

Committee in charge:

Ryan Charles Kastner, Chair
Pamela Cosman
Zhaowei Liu
Manuela Vasconcelos

2016

The thesis of Nirja Jatin Mehta is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____

Chair

University of California, San Diego

2016

DEDICATION

I would like to dedicate this thesis to my parents and my sisters. This thesis would not have been possible without their endless love and support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my parents - Dr. Dipti Mehta and Dr. Jatin Mehta, my sisters - Dr. Shrija Shah and Nupur Mehta, and my dear friend - Aashil Togadia for their constant encouragement and patience. Their faith in me has been of utmost importance prior to and during the course of my graduate program.

I would like to express my sincere gratitude to my advisor Professor Ryan Charles Kastner for his invaluable advice and continuous support. His drive, attitude and principles on research has provided a precious lesson throughout my work on this thesis. His immense knowledge and guidance helped me in conducting research.

Besides my advisor, I would like to thank my thesis committee members, Professor Zhaowei Liu, Professor Pamela Cosman and Professor Manuela Vasconcelos, for their time to review my research and for their valuable comments.

I would like to thank Dajung Lee for her immense help and support in conducting research, reviewing this thesis and providing insightful comments. I would also like to thank her for being my partner in the initial stages of this work and working tirelessly towards co-authoring a paper under review. I am grateful to Dajung, Ryan and Alexandria for co-authoring the following paper with me. Lee, Dajung; Mehta, Nirja; Shearer, Alexandria; Kastner, Ryan “A Hardware Accelerated System for High Throughput Cellular Image Analysis,” Journal of Parallel and Distributed Computing, submitted.

VITA

1991	Born, Ahmedabad, India
2013	B.Tech., Electronics and Communication Engineering, Institute of Technology, Nirma University, Gandhinagar, India
2014	Software Engineer, Cybage Software Pvt. Ltd., Gandhinagar, India
2016	M.S., Electrical Engineering (Signal and Image Processing), University of California, San Diego

ABSTRACT OF THE THESIS

Automated Cell Detection and Morphology Analysis on Microscopic Images in Imaging Flow Cytometry

by

Nirja Jatin Mehta

Master of Science in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2016

Ryan Charles Kastner, Chair

Recent advances in imaging technologies have eased the collection of microscopic images, but efficient image analysis of this data remains a challenge. In the field of imaging flow cytometry, accuracy, simplicity and processing time of image processing algorithms plays a significant role. The microscopic image data acquired by video camera is of low contrast and low resolution. In addition, the camera film may be subject to varying light levels. Hence, it is crucial to develop image analysis techniques that are independent of image quality and inconsistent light levels, consequently enabling automation of their analyses. This thesis studies and implement various image analysis techniques on different subpopulations of cell that are subject to hydrodynamic pressure in microfluidic medium. It focuses on examining the credibility of these techniques by comparing against manually collected ground data. Our study targets pixel intensity-based thresholding techniques in single cell detection, cell area calculation and nucleus

isolation. The studies in this thesis also indicate the advantages of implementing simpler image morphology algorithms for collecting cell morphology features used in imaging flow cytometry applications.

Chapter 1

Introduction

In recent years, characteristics of cells flowing along a microfluidic channel have been studied to extract information about its malignancy state and abnormalities. Measurements of the cell and its organelles is an accurate predictor of any associated abnormalities [1]. The study of properties of cellular specimen suspended in a fluidic medium is called flow cytometry. The images acquired by scanning a cellular specimen are further analyzed to perform image cytometry, i.e. the measurement of individual cells. A unique combination of image analysis and microscopy yields cellular measurements that can be used for further classification of cells.

With this thesis, we discuss many image processing methods to account for image separation of a particular cellular specimen. We aim to develop algorithms that facilitate complete automation of imaging flow cytometry which will help in practical analysis of large number of cells. Therefore, developing a system capable of unattended analysis of cells and related morphological features.

1.1 Motivation

The number of microscopic images analyzed by methods such as Coulter Counter [2] is limited. Also, some of the standard microscopic methods require significant human interaction and are thus quite slow. Ortyn et al. [3] in their work, state that manual categorization of a few hundred cells on the basis of morphological features result in poor statistical results in evaluating observed changes over time across the cellular specimen. It would be beneficial to implement automated image analysis methods that identify cellular abnormalities or cellular distribution abnormalities associated with a disease condition.

Imaging flow cytometry combines the high-throughput potential of conventional flow cytometry with single-cell imaging. This method has proven to be efficient in conducting cellular image analysis. One of the widely used methods is fluorescence microscopy. It uses spectral information from the image dataset as an indicator of cellular characteristics like shape, size, location to name a few. However, in this work, the idea is to report quantitative image analysis of bright-field microscopic images collected by imaging flow cytometers that enables imaging cellular specimen without the use of fluorescence biomarkers. Avoiding fluorescent stains reduces effort and cost, and avoids potentially confounding side effects [6] of live cell markers. Imaging flow cytometry capable of fully automating the process of cellular analyses is necessary for high-throughput analyses systems. This technique would yield accurate results if the cytometer had a stable source of light. In absence of steady illumination, it is imperative to develop image analysis algorithms that are independent of varying light levels.

1.2 Our Contributions

We present a high-throughput image analysis strategy for a flow cytometry application. This is a label-free imaging technique that accounts for varying light levels of the acquired microscopic images. It is not carried out by using fluorescence microscopy or any other reagents. In absence of fluorescence image data, it is difficult to analyze the image as the variation of intensities is only in the pixel intensity range of 0-255 gray values. With this thesis, we design, implement and assess the performance of different filtering algorithms aimed at cell detection and cell morphology of bright-field grayscale microscopic images.

Recent advances in imaging flow cytometry show promising potential in the field of clinical diagnostics, drug discovery and biological research [4]. Intuitive image-based results are facilitated by high speed image acquisition coupled with accurate image analysis. The image sets obtained from flow cytometers are huge in number and smaller in size. To match up to the image acquisition speed, we propose our image analysis algorithms that use simple methods like manipulation pixel intensities for cell detection and cell morphology features.

In this work, we present our strategy for high-throughput image analysis for cell detection, tracing and analyzing cell morphology in high-speed images. We have been successful in achieving 80% sensitivity, 99.53% specificity, 96.43% precision and 96.9% accuracy for one of the image histogram-based cell detection methods. Computation of cell area was also shown to be correlating with ground truth numbers for hysteresis thresholding method on test dataset containing bladder epithelial cells and principal component analysis-based thresholding on test

dataset containing white blood cells.

1.3 Limitations

The image analysis algorithms designed and implemented in this thesis are aimed at examining microscopic images with a single cell. This work does not consider the analysis of microscopic images with multiple cells.

Our cell detection algorithms set a specific cut-off limit to detect the presence of a cell in the last stage of cell detection process according to the algorithm in use. While this constraint is somewhat arbitrary, it is reasonable given that each algorithm produces a different thresholding point that results in different binary images.

Cell detection algorithms are implemented on MATLAB R2015b and the calculations of cell area and nucleus isolation are performed on Python 2.7.

The thesis accounts for detection of cells of smaller sizes with varying light-levels while cell area and nucleus isolation is analyzed on comparatively larger cell sizes. While we understand the inconsistency due to this discrepancy, we designed the algorithms based on the test dataset that we were provided with.

1.4 Outline of the Thesis

The rest of the thesis is organized as follows: In Chapter 2, we review techniques used by previous works towards image analysis algorithms in low resolution, low quality microscopic images.

In Chapter 3, we discuss the design and implementation of different cell detection algorithms. These algorithms were developed and tested on MATLAB R2015b.

In Chapter 4, we discuss the design and implementation of algorithms for calculation of cell area. These algorithms were developed and tested on Python 2.7.

In Chapter 5, we discuss the methodology to isolate the cell nucleus region.

In Chapter 6, we present quantitative results of performance of different algorithms described in previous chapters against the collected ground truth data, along with directions for future work.

Chapter 2

Background and Previous Works

In recent years, a combination of fluorescence microscopy and flow cytometry has been successful in analyzing cellular images that are captured across the microfluidic channel. While cellular specimen stained with fluorescent dyes are widely used for further image analyses, Darling et al. [8] proposed a novel method of assessing mechanical properties of cells as a health assessment metric. With this thesis, we discuss different image processing techniques post-microscopic image acquisition. We propose different solutions for three types of experimental image sets that contain white blood cells and bladder epithelial cell samples. We report quantitative image analysis results for cell detection and cell morphology features.

2.1 Background

Manual inspection of tissue and blood films from patients is a time-consuming task and also prone to human error. The changes in morphological appearance of cells under study over time are difficult to detect by human eye. Human observers can point fewer cellular features as compared to imaging flow cytometry that yields many informative measures of cells. Human-scored image analysis is qualitative but on the other hand, automated analysis provides consistent and quantitative measure for every image [5]. Automated image analysis is useful in measuring large number of features that are not detectable even by human eye and hence has proven useful for cytometry applications. Imaging flow cytometry is much less labor-intensive and yields higher-throughput results.

While imaging techniques give successful results in high-throughput image acquisition rates, a bottleneck exists at the image analysis stage [5]. For most applications, automated cell

analysis is required to extract the full spectrum of information present in biological images. It is therefore required to develop an automated cell analysis method to analyze imagery to identify cellular abnormalities or cellular distribution abnormalities associated with a disease condition [3]. Image cytometry would therefore, relieve the stress of routinely conducting biological experiments.

One of the most widely imaging techniques used is fluorescence microscopy. In this technique, multimodal or multispectral images of cells comprising a population of cells are simultaneously collected. The changes in stained-cells are studied as they are proven to be indicative of a disease state and also indicative of relative effectiveness of a therapy the cellular specimen is subject to. However, measurements of cellular mechanical properties have been in focus of many cytometric studies [1] as these numbers are associated with malignancies and abnormalities. Gossett et al. [7] mention that the use of intrinsic biophysical markers such as the ability to mechanically deform under a load have gained momentum in the field of clinical diagnostics rather than the usage of biochemical markers followed by costly labeling process or sample preparation.

2.1.1 Advantages of Studying Cellular Mechanical Properties

Mechanical phenotyping of cells is primarily used to differentiate between healthy and unhealthy cells [8]. Studying cellular mechanical properties of single cells can provide clear indications of the status and progression of the disease. Monitoring the mechanical properties of such cells may serve as a predictor in determining whether a patient can effectively fight infection. Darling et al. [8] state that morphological features of cells have been used for identifying and staging malignancy, and analyzing these features in cells from body fluids and biopsies is routine in cytology and pathology labs. High-throughput approaches to mechanical phenotyping, coupled with improved sensitivity technology is the need of the hour. In addition to that, the automated diagnosis of cell samples should be a fast process and provide quantitative, comparable test results.

Cancer is a prominent target of investigation [21]. Researchers have repeatedly observed that malignant cells tend to be softer under mechanical pressure than their healthy counterparts. Remmerbach et al. [10] found that oral cancer cells were 3.5 times more compliant than healthy cells.

Fedosov et al. [11] state that when Red Blood Cells (RBCs) are infected with the parasite Plasmodium falciparum, they lose their deformability, with cell stiffness increasing by more than

10-fold [22]. Osteoarthritis is another representative disease that changes the mechanical phenotype of unhealthy cells [8]. The cellular mechanical properties are affected by the aging process as well resulting in degradation of cellular function [12], [13]. For most cell types, age causes an increase in cell stiffness along with an inability to fully recover from large deformations. Hence, studying the change in cellular mechanical properties would act as an advantageous metric in the process of health assessment of cell.

2.2 Previous Works

Coulter counter and flow cytometers are one of the well-known devices used for semi or fully automated analysis of cell. The coulter counter provides information about the cell number by its cell counting principle. On the other hand, the flow cytometer is a fluorescence activated cell sorter (FACS), that uses fluorescence reagents to highlight and measure cells suspended in a fluidic medium. The flow cytometer gives information about cell number, size and number of cellular substances labeled with the fluorescence dye. The number of microscopic images analyzed by these methods is limited. Also, some of the standard microscopic method require significant human interaction and are thus quite slow.

Imaging flow cytometry integrates the high-throughput capabilities of conventional flow cytometry with image cytometry of single cell. In addition, measurements of individual cells in situ have been performed with varying degrees of automation in microscopy [1]. Cell state, in general, is studied under the effect of biochemical and biophysical markers. In flow cytometry, the cellular specimen is stained by dye or fluorescence reagents. Multispectral analysis is performed on the acquired image data to measure morphological features of the cells. In [1], Price et al. suggest an image processor that applies fourier-space analysis along with an iterative thresholding technique on the acquired images. Image segmentation involves filtering of the digital image containing a cell followed by thresholding the resultant image [1].

Cell measurement has evolved in the recent years and is implemented in many commercially available cytometers. Table 2.1 reviews the capability of such imaging flow cytometers according to the imaging modality used and the measurement information obtained.

All devices/software except CellProfiler mentioned in Table 2.1 use the same imaging modality i.e. using fluorescence dyes. CellProfiler [5], [6] is one of the only open-source biological image analysis softwares capable of working with bright-field microscopy images. Its design, however is aimed towards dataset containing multiple human cells in a single image. In addition, CellProfiler's object identification modules contain a variety of published and tested

Table 2.1: Comparison of Imaging Flow Cytometers

Application	Imaging Modality	Measurement
ThorLabs iCyte Automated Imaging Cytometer	Fluorescence, chromatic dyes	Cell area, perimeter, circularity, location
Amnis ImageStreamX Mark II Imaging Flow Cytometer	Fluorescence	Cell circularity, cell signalling
GE Cytell Cell Imaging System	Fluorescence	Cell count, concentration, cell cycle phase distribution, cell viability
Broad Institute CellProfiler	Bright-field, dark-field, fluorescence	Cell, size, shape, intensity and texture

algorithms for identifying cells based on fluorescence [5]. Our work, on the contrary, focuses on single-cell bright-field image cytometry in order to aim for high-throughput image acquisition and image analysis rate. The image data does not contain any spectral content and hence it is a challenging task to apply image processing algorithms on this type of image data.

2.3 Image Acquisition Method

Research about accurately measuring cellular mechanical properties has been ongoing since many decades. A variety of tools have been used to examine the deformation response of materials at the microscopic level [14]. In their work, Darling et al. [8] propose hydrodynamic approaches that include a microfluidic channel along which the cells are subjected to intrinsic fluid-dynamic stresses.

2.3.1 Experimental Setup

Parallel, and precise cell and particle manipulation can be achieved using inertial lift forces intrinsic to particle motion in confined channel flows [15] which is coined as inertial focusing. Biological fluids and cellular suspensions are processed using this technique. A unique combination of inertial focusing, hydrodynamic stretching, and automated image analysis enabled Lee et al. [9] to carry out tunable single-cell mechanical measurements of cells with a throughput several orders of magnitude greater than current systems and is capable of probing

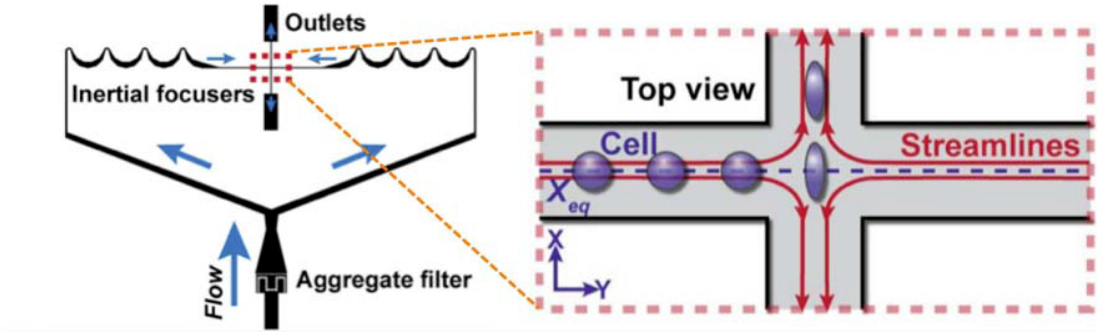

Figure 2.1: Experimental setup for imaging flow cytometry [9].

single-cell deformability at approximately 2,000 cellss. The method uses inertial focusing to uniformly deliver cells to a stretching extensional flow where cells are deformed at high strain rates, imaged with a high-speed camera, and computationally analyzed to extract quantitative parameters.

The experimental setup by Lee et al. [9] for deformability cytometry is shown in Figure 2.1.

A solution containing cells enters from a single inlet as shown. It separates and flows in two directions, that proceeds into an inertial focusing channel. The inertial focusing [16], [17] channel is designated to center the cells as they enter the extensional flow region. Cells can enter this region from either of the two directions. Convergence of the two channels creates a hydrodynamic strain on the cells in this region. The cells are surrounded by fluid and do not come in contact with each other. Figure 2.2 shows a sequence of image frames as a cell moves through the extensional flow region.

The cell is moving from left to right in the channel. As it reaches the center of the channel, force from the two merging flows causes the cell to deform and it is this deformation we wish to measure. This information can be used to characterize and sort the cell. Cells are analyzed based on morphological parameters, such as their initial diameter, circularity, and deformability.

The change in aspect ratio of the cell is a good metric to measure the circularity and deformability. For that the exact radius at each point of the cell membrane should be known. Darling et al. [8] suggest radius measurement by bisecting the cell and then radially sweeping the bisection plane around 360 degree like a clock hand. An example of the output is shown on the right side of Figure 2.2.

By using deformability cytometry, sampling thousands instead of tens of cells is possible

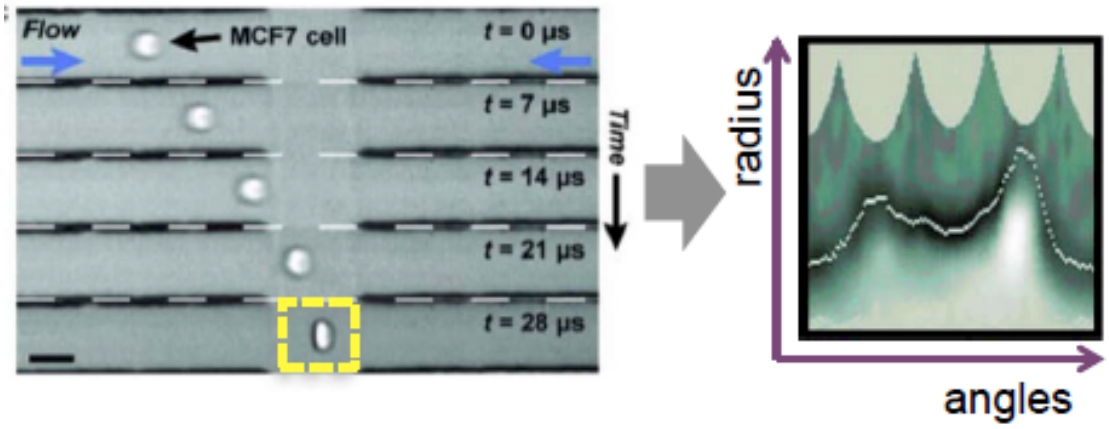

Figure 2.2: Automated image analysis tracks the cell through the extensional flow region (left) and determines morphological properties, derived from the radius at each angle (right) [9].

such that all cells in a complex population can be measured without preselection [19], [20]. The image data collected after this type of microscopy procedure is used as the data source for designing and implementing cell detection and tracing algorithms. Cell morphology characteristics such as area of cell and isolating cell nuclei are studied on a different image dataset containing white blood cells and bladder epithelial cells.

2.3.2 Image Acquisition Format

Different sensor technologies are used to capture biomedical images which are digital images in terms of visual representation. The images captured by Amnis ImageStream system are gray-scale images with light intensities represented by constant corresponding to the brightness of a pixel at coordinates (x,y) called the gray-level. It is notable that wider the range of the gray level, better is the achieved resolution [23].

.bin Files

This section describes the cine (.cin) file format used for saving video information and auxiliary data captured from digital video cameras. Cine file format was designed for storing and retrieving the recordings made by the Phantom high speed video cameras from Vision Research Inc.

The cine format describes the image data that comes from Phantom cameras. The cine file acquired from this camera is used to store both the data about the pixels of images and the additional information like the acquisition parameters, image time, analog signals recorded

in parallel with the images, range data, etc. A cine file contains all the data produced at the recording of an event so one can retrieve information, playback the images, analyze them, etc. The pixel information can be raw interpolated uncompressed or compressed. The raw format is preferred because it is fast to save, the file has smaller size and there is not any loss of information at save. Cine files are compressed to files containing only image data and extracting the metadata from acquired images which are called .bin files. For the purpose of this thesis, .bin files are used to analyze cellular image data and implement cell detection algorithms.

I am grateful to Dajung, Ryan and Alexandria for co-authoring the following paper with me. Lee, Dajung; Mehta, Nirja; Shearer, Alexandria; Kastner, Ryan “A Hardware Accelerated System for High Throughput Cellular Image Analysis,” Journal of Parallel and Distributed Computing, submitted.

Chapter 3

Cell Detection Methods for Imaging Flow Cytometry

In the present chapter, we discuss different techniques to detect the presence of a cell flowing through a microfluidic channel using a morphological image analysis based method. Accurate cell detection in an image is the first step towards fully automated measurement of cell characteristics. The automated analysis script was built in MATLAB R2015b (MathWorks, Natick, MA). The algorithm examines high-speed bright-field microscopic images (64x64 pixels) of cells in a high-throughput flow through the system. The algorithm (1) performs image adjustments, (2) detects the presence of an object, (3) tracks its motion, and (4) performs a measurement of morphology.

For this purpose, an imaging system is set up as described in Chapter 2 and the images were captured by a Phantom high-speed digital video camera. The cell images captured are at different illumination levels due to varying room illumination and extra light sources. Such differences can yield misleading results when the images are analyzed and compared with each other [23]. For instance, the image processing algorithm that works on images of one dataset may not work effectively on the other. This arises the need for an automated image processing algorithm. Morphological image processing methods have been developed in a MATLAB environment, to detect the presence of different cells along the microfluidic channel under varying illumination levels. The core idea is to pre-process the raw image frame, threshold it and apply morphological operations to detect the cell against background. Common image segmentation methods include thresholding, edge detection and region extraction. For precise edge detection, the variation of pixel intensities in an image must be well-defined. The images in our test dataset

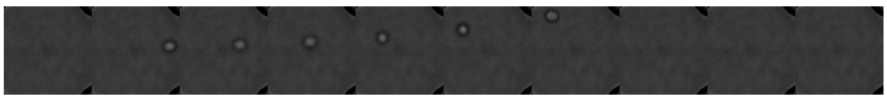

Figure 3.1: Stream of 10 frames of b.bin file.

are of low contrast and therefore, edge detection methods would not be beneficial. Region extraction methods are iterative and hence can be computationally expensive. Hence, different thresholding techniques are implemented on the image datasets and described in detail in this section.

The provided video files are in .bin format. Each file contains nearly 5,000 image frames. Each microscopy image under study is a 64x64 pixels gray-scale image. Typically, each frame will either contain no cell or only one cell. However, there are cases where one cell is one or two frames away from exiting while another cell starts entering in and hence, such frames contain two cells. Algorithms involving morphological operations are restricted to detect the presence of a single cell. A stream of 10 image frames is shown in Figure 3.1.

To detect the presence of a cell, the first step is to find the background for each test dataset. The image frame under study rejects the background and is processed in different ways for each of the following methods.

We started with an initial image analysis algorithm for cell detection, namely given method (refer to Chapter 6 for results). The block diagram of this method is shown in Figure 3.2. In this method, the original image is subtracted from an average of first 10 frames.This subtracted image is then binarized against a constant threshold. The total number of white pixels in the binary image is counted. If the count is more than a set number, the cell is pronounced to be present.

3.1 Method 1 - Contrast Stretch Method

Each set of pixel intensity ranges can be used to separate the background from cellular region if the intensities were well-defined. Because of low contrast of the images, simple thresholding techniques would not yield accurate results. To subtract the background, a background frame for each image frame in a test dataset is calculated as the average of 10 image frames of a single test dataset. The averaged frame is shown in Figure 3.3 for b.bin file.

As seen from Figure 3.1 and Figure 3.3, the image frames contain low contrast and hence it becomes tricky to apply image processing techniques on a low-contrast image.

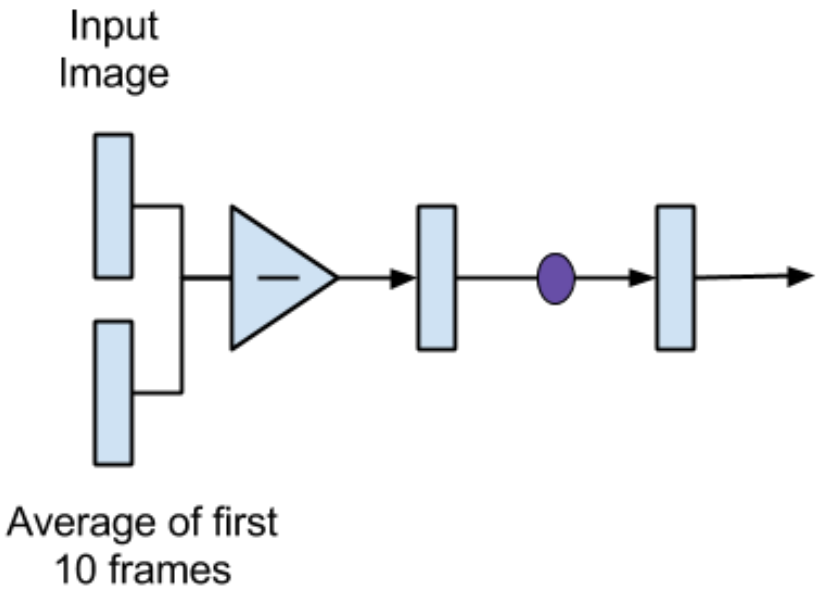

Figure 3.2: A block diagram of given method.

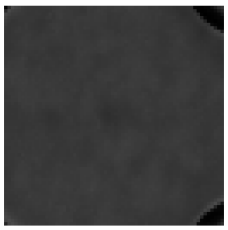

Figure 3.3: Background frame for contrast stretch method.

3.1.1 Point Processing

Point processing consists of a mathematical transformation on the image pixels that modifies the original pixel values to create the values of corresponding pixels in the enhanced image [23]. Mathematical expression for such a transformation is described in Equation 3.1.

$$g(x, y) = T[f(x, y)] \tag{3.1}$$

Where *f(x,y)* is the original image, *g(x,y)* is the transformed image, and T is the transformation function between these two images. The pixel values at a point (x,y) in spatial domain in the original image is denoted by *r(x,y)* and the pixel values in the transformed image are denoted by *s(x,y)*. The mathematical expression in terms of pixel values is given by Equation 3.2.

$$s(x, y) = T[r(x, y)] \tag{3.2}$$

Different gray-level transformations include negative transform, logarithmic transform, inverse-log transform, and power-law transforms. Choice of the transformation technique is dependent on the objective of the point processing task. The purpose of point processing is to improve the quality of the image by expanding the gray-level range [23]. The object of interest, in this case, the cell occupies only a specific range of gray values. By manipulating the image, the object of interest occupies a larger range of the gray level [23] and hence it is easier to extract information out of the image for future processing like thresholding. Contrast enhancement is a method to create images with better visibility by stretching the gray-scale range.

Logarithmic Transformation

To enhance the image contrast, we use the method of log-transform contrast stretch. The mathematical operation to map gray-values from original image onto the transformed image is given by Equation 3.3.

$$s(x, y) = c * log(1 + r(x, y)) \tag{3.3}$$

The logarithmic transformation maps a narrow range of low input gray-values into a wider range of output values [25]. Maini et al. [24] state that logarithmic transformation is effective to enhance details in the darker regions of the image at expense of detail in the brighter regions the higher-level values. For hardware simplicity, a look-up table is calculated that maps gray-value of input pixel with the value of output pixel. The stretch of gray-level range is shown

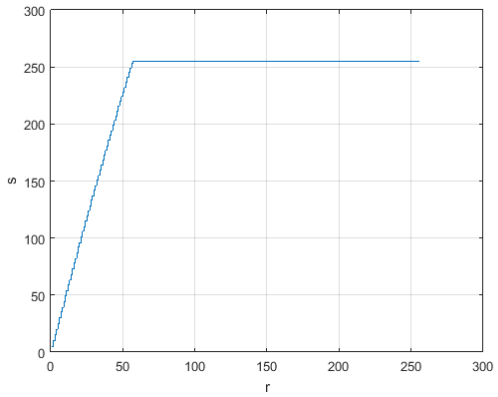

Figure 3.4: Logarithmic transformation.

in Figure 3.4. From Figure 3.1 one can also notice how each image frame contains the black and white stripes at each of the four corners. Logarithmic transformation on these image frames also enhances these stripes, which are of no interest for cell detection. Hence, these stripes are set as background pixels. However, a cell at either of the four corners will not be detected because of this explicit backgrounding.

3.1.2 Thresholding Technique

Thresholding, also referred to as gray-level segmentation is a conversion technique between gray-level image and a bi-level image (binary image) [26]. An ideal thresholding technique would yield an image containing areas of interest in an image (in our case, the cellular region) and hence helps in reducing the complexity of the image data for future processing tasks like erosion and dilation. The most common thresholding technique is selecting a constant value as the threshold. All gray-values below this constant will be classified as black pixels and all those gray-values above this constant will be classified as white pixels in the binary image. This type of thresholding is only true in some cases because of noise and illumination effects [26]. However, the input image for thresholding is a contrast stretched image and is free from any effect due to varying light levels.

The enhanced image is compared against a set threshold (constant) value. This threshold is determined from the pixel value found in the image and selected by trial-and-error. If there is a significant number of white pixels, the cell is pronounced to be present. To illustrate the implementation of contrast stretch method, its block diagram is shown in Figure 3.5. The cell detection part is highlighted in purple.

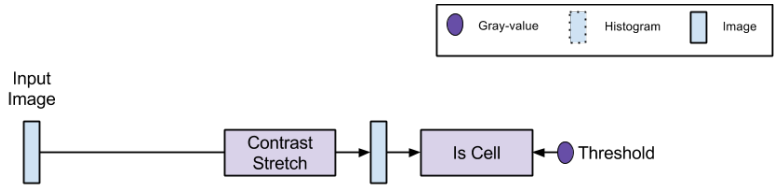

Figure 3.5: A block diagram of contrast stretch method.

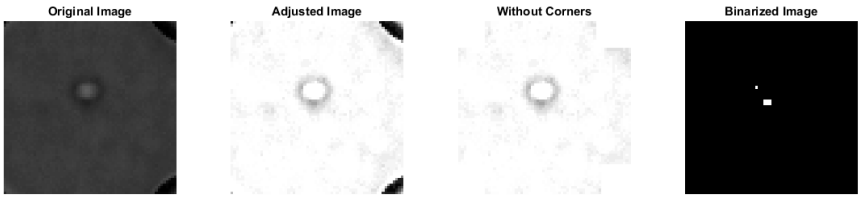

Figure 3.6: Contrast stretch - cell detection flow.

Step-wise implementation of the above mentioned calculations is shown in Figure 3.6.

3.2 Method 2 - Histogram-based Thresholding 1 (HBT I)

A stream of 10 image frames is shown in Figure 3.1 in which the cell enters around second image frame and exits in seventh image frame.

As seen in Figure 3.1, the contrast of all images is low but it does not vary across all frames. Also, the background appears like a texture which is constant throughout the stream of image frames. This property is used as a motivation for deciding the background frame. The background is chosen as the image frame where cell is not present, instead of taking an averaged image across 10 frames. The background frame is shown in Figure 3.7 for b.bin file. The background frame is subtracted from each image frame under study.

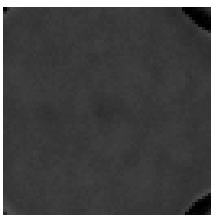

Figure 3.7: Background frame for HBT I.

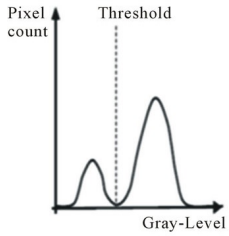

Figure 3.8: Optimal threshold selection by Otsu's method.

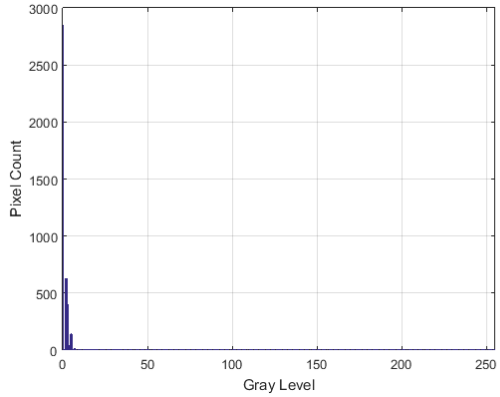

Figure 3.9: Histogram of background subtracted image.

3.2.1 Thresholding Technique

The background subtracted image is thresholded against a constant. This constant value is calculated based on the histogram of resultant image. The histogram is used for calculating intensity statistics. One common assumption is that the foreground and background pixels are contained under a Gaussian curve resulting in a bi-modal distribution as shown in Figure 3.8. The optimal threshold value to separate the foreground from background is the low point between the two peaks in the histogram. However, for most biomedical images it is difficult to find the low point [27] as the image is imbued with noise and low contrast. One of the widely used thresholding technique is a bi-level thresholding method called Otsu's method which is based in minimizing the within-class variance. Pixels of image are divided in two classes based on a threshold value t. Otsu's method chooses the optimal threshold t by maximizing the between-class variance and hence minimizing the within-class variance [28]. A typical example of gray-level thresholding by Otsu's method is shown in Figure 3.8. This method consists of an unsupervised nature of threshold selection but has certain limitations.

Figure 3.9 shows a typical histogram of background subtracted image.

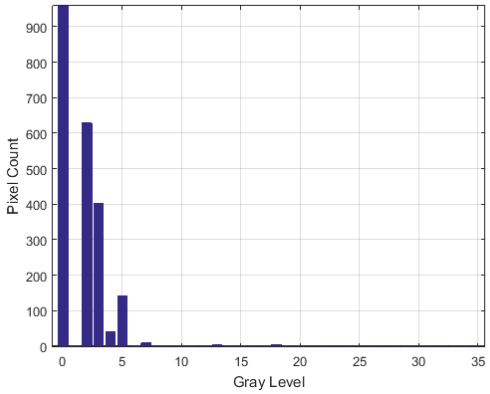

Figure 3.10: Zoomed in histogram of background subtracted image.

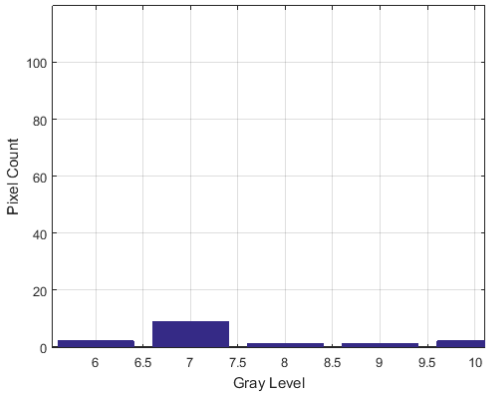

Figure 3.11: Closer look at histogram of background subtracted image.

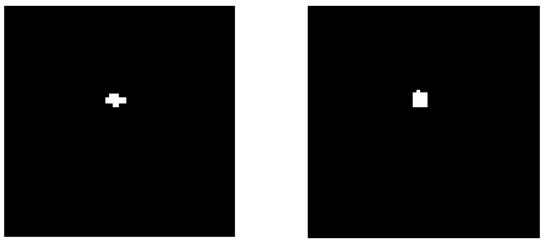

Figure 3.12: Set of eroded images from HBT I method.

Figure and 3.9 and 3.10 shows that the background subtracted image contains large number of dark pixels while the gray-values corresponding to the cell area is in the gray-level range of 5-10. Otsu's method would yield a threshold value of 1 rather than a value between 5-10. Applying a threshold of 1 to the resultant image would take into account background pixels along with the cell and hence give undesired results. As seen in Figure 3.10 and 3.11, the gray-values corresponding to the cell occur for pixel counts much lesser than 500. After calculating the histogram, threshold value is selected as the one corresponding to small number of pixel counts as the cell occupies a petite region in the image. Threshold value is set as the value calculated from the first frame that contains a cell. This value is set constant throughout rest of the file.

3.2.2 Image Opening

Morphological operations on a binary image are widely used to find the shape of an organism which can be used in the process of cell sorting. The binary image contains some isolated background pixels that need to be removed for further processing. This process is carried out by binary erosion operation. To enhance the shape of the cell, dilation is applied to the eroded image. The combined process of binary erosion and dilation is called image opening. This application tends to open small gaps or spaces between touching objects in an image [26]. After opening the binary image, if the cell is present, it is isolated and is ready for further processing.

Two structuring elements each, for binary erosion and dilation are used in this implementation. For erosion, the structuring elements used are a 3x1 and a 1x3 array of white pixels. The eroded image for one image frame is shown in Figure 3.12. The eroded images are dilated by structuring elements 6x5 array and 5x6 array of white pixels. The opened image is then formed by OR-ing both dilated images. The dilated images are shown in Figure 3.13.

If there is a significant number of white pixels, the cell is pronounced to be present. To illustrate the implementation of HBT I method, its block diagram is shown in Figure 3.14. The

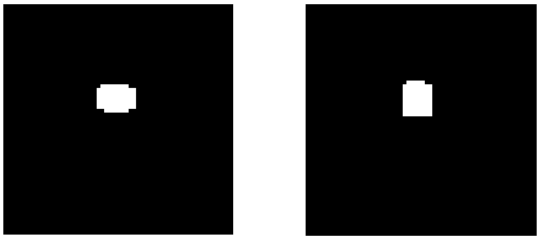

Figure 3.13: Set of dilated images from HBT I method.

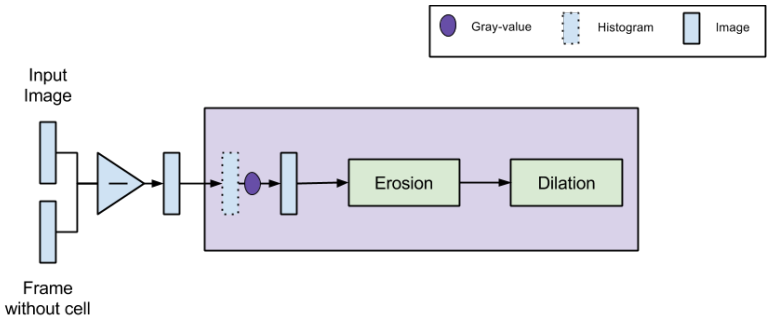

Figure 3.14: A block diagram of HBT I method.

cell detection part is highlighted in purple.

Step-wise implementation of the above mentioned calculations is shown in Figure 3.15.

3.3 Method 3 - Histogram-based Thresholding 2 (HBT II)

The background frame is same as in the case of HBT I method.

3.3.1 Thresholding Technique

The differentiating factor is the way of calculating the threshold value. Calculating the histogram for each image is a computationally expensive procedure. Instead, the threshold value

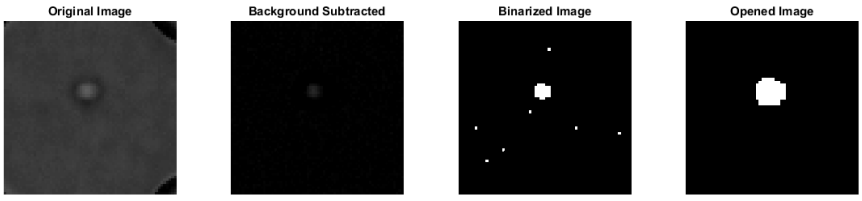

Figure 3.15: HBT I method - cell detection flow.

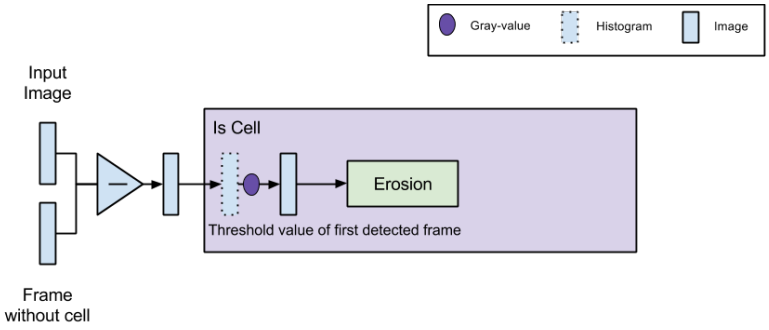

Figure 3.16: A block diagram of HBT II method.

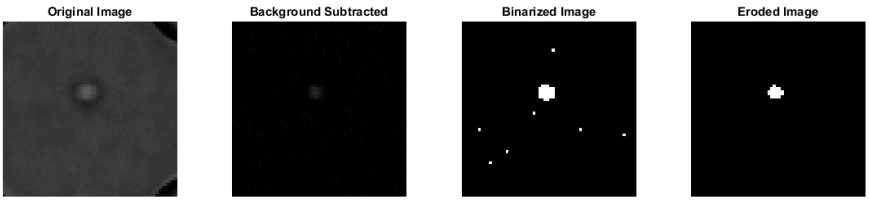

Figure 3.17: HBT II method - cell detection flow.

is set as the value of first frame that contains a cell. Once a cell is detected, the threshold value is noted and set equal to the threshold value throughout the test dataset being processed.

3.3.2 Image Erosion

The binary image contains some noisy pixels around the cell. To remove it, morphological operations like binary erosion and dilation are used. On close inspection, we could observe that to find the presence of a cell, binary erosion alone is effective rather than erosion and dilation combined. For binary erosion, the structuring elements used are a 3x1 and a 1x3 array of white pixels.

If there is a significant number of white pixels, the cell is pronounced to be present. To illustrate the implementation of HBT II method, its block diagram is shown in Figure 3.16 The cell detection part is highlighted in purple.

Step-wise implementation of the above mentioned calculations is shown in Figure 3.17.

3.4 Method 4 - Minimum Error Thresholding

The background frame is same as in the case of HBT I method. The differentiating factor is the way of calculating the threshold value.

3.4.1 Thresholding Technique

Apart from direct selection of the threshold, the determination of a suitable threshold involves the computation of the histogram of gray-values and its subsequent analysis [28]. The histogram of image can be thought of as a measured probability density function of the two distributions (foreground pixels and background pixels) [26]. The distribution is assumed to be under a pair of Gaussian curves with distinct means and standard deviations. The probability density function p(g) of the mixture population containing the foreground and background pixels is given by [30] in Equation 3.4.

$$p(g) = \frac{1}{\sigma_1\sqrt{2\pi}}e^{-}(\frac{(g-\mu_1)^2}{2\sigma_1^2}) + \frac{1}{\sigma_2\sqrt{2\pi}}e^{-}(\frac{(g-\mu_2)^2}{2\sigma_2^2}) \tag{3.4}$$

Where σ_1 and μ_1 are the standard deviation and mean of either of the classes, and σ_2 and μ_2 are the standard deviation and mean of the other class. Taking logarithm of both sides and re-arranging, we get a quadratic equation that could be solved for g [26] as shown in Equation 3.5.

$$\frac{(g-\mu_1)^2}{2\sigma_1^2} + log\sigma_1 - 2logP_1 = \frac{(g-\mu_2)^2}{2\sigma_2^2} + log\sigma_2 - 2logP_2 \tag{3.5}$$

Where P_1 is the priori probability for one of the classes and P_2 is the priori probability for the other class. The values of μ, σ and P are not known and can be estimated by the criterion function proposed by Kittler and Illingworth [26], [30] as shown in Equation 3.6.

$$J(t) = 1 + 2(P_1(t)log\sigma_1(t) + P_2(t)log\sigma_2(t)) - 2(P_1(t)logP_1(t) + P_2(t)logP_2(t)) \tag{3.6}$$

where

$$P_1(t) = \sum_{g=0}^{t} h(g) \tag{3.7}$$

$$P_2(t) = \sum_{g=t+1}^{255} h(g) \tag{3.8}$$

$$\mu_1(t) = \frac{\sum_{g=0}^{t} g.h(g)}{P_1(t)} \tag{3.9}$$

$$\mu_2(t) = \frac{\sum_{g=t+1}^{255} g.h(g)}{P_2(t)} \tag{3.10}$$

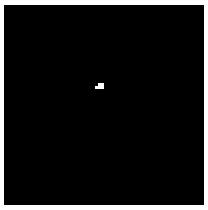

Figure 3.18: An eroded image from minimum error thresholding method.

$$\sigma_1^2(t) = \frac{\sum_{g=0}^{t} h(g)(g - \mu_1(t))^2}{P_1(t)} \tag{3.11}$$

$$\sigma_2^2(t) = \frac{\sum_{g=t+1}^{255} h(g)(g - \mu_2(t))^2}{P_2(t)} \tag{3.12}$$

The value of threshold *t*, that accounts for minimum error is calculated by the value that minimizes J(t). This method is referred to as minimum error thresholding [26].

3.4.2 Image Erosion

Morphological operations like binary erosion and dilation are also computationally expensive. On close inspection, we could observe that to find the presence of a cell, erosion alone is effective rather than erosion and dilation combined. The thresholded image contains a large chunk of the cell along with isolated background pixels. The background pixels are removed by a disk structuring element of radius 2. The resultant eroded image for one image frame is shown in Figure 3.18.

If there are one or more number of white elements in the eroded image, a cell is pronounced to be present. To illustrate the implementation of minimum error thresholding method, its block diagram is shown in Figure 3.19.

Step-wise implementation of the above mentioned calculations is shown in Figure 3.20.

I am grateful to Dajung, Ryan and Alexandria for co-authoring the following paper with me. Lee, Dajung; Mehta, Nirja; Shearer, Alexandria; Kastner, Ryan "A Hardware Accelerated System for High Throughput Cellular Image Analysis," Journal of Parallel and Distributed Computing, submitted.

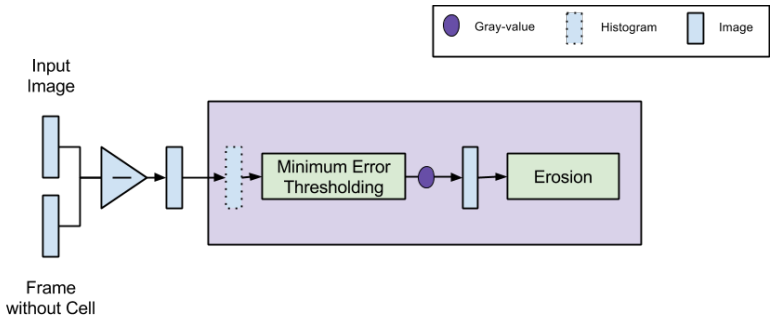

Figure 3.19: A block diagram of minimum error thresholding method.

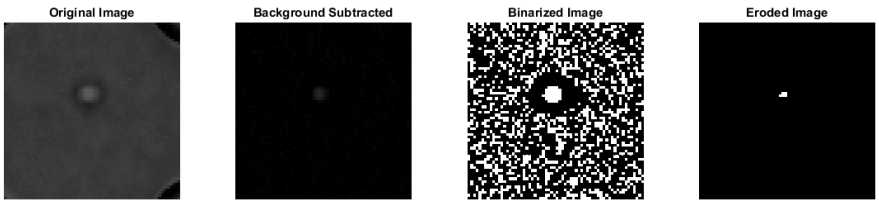

Figure 3.20: minimum error thresholding method - cell detection flow.

Chapter 4

Algorithms to Calculate Cell Area

In the present chapter, we discuss the assessment of cellular area of bladder epithelial cells and White Blood Cells (WBCs) flowing through a microfludic channel using an image analysis based method. For this purpose, the images were captured by a microscopy system. Automated thresholding, clustering and segmentation methods have been developed in a Python environment to calculate the area occupied by both healthy and exposed cells.

In case of automated thresholding techniques, the idea is to detect the cell based on different gray-values and selecting a threshold above which the pixel intensity matches the cell region. In case of segmentation techniques, the idea is to collect similar pixels (cell) based on a particular metric to form representations that emphasize important, suggestive and distinctive properties of the cell [35]. Clustering algorithms use distance as one metric in finding the similarity among points. To detect the cell area, we implement unsupervised image clustering and segmentation techniques like k-means.

A stream of image frames containing a cell is termed as a cell event in which each image frame contains a cell. The cell could be paired with the membrane and a nucleus. Each image frame is 128x128 pixels in resolution. A typical cell event is shown in Figure 4.1.

The image frames contain low contrast and hence needs to be pre-processed. The image is conditioned by normalizing the intensity values and taking the absolute values. To reduce the effect of the noisy background, the pre-processed image frame is then smoothened by a Gaussian

Figure 4.1: An example cell event.

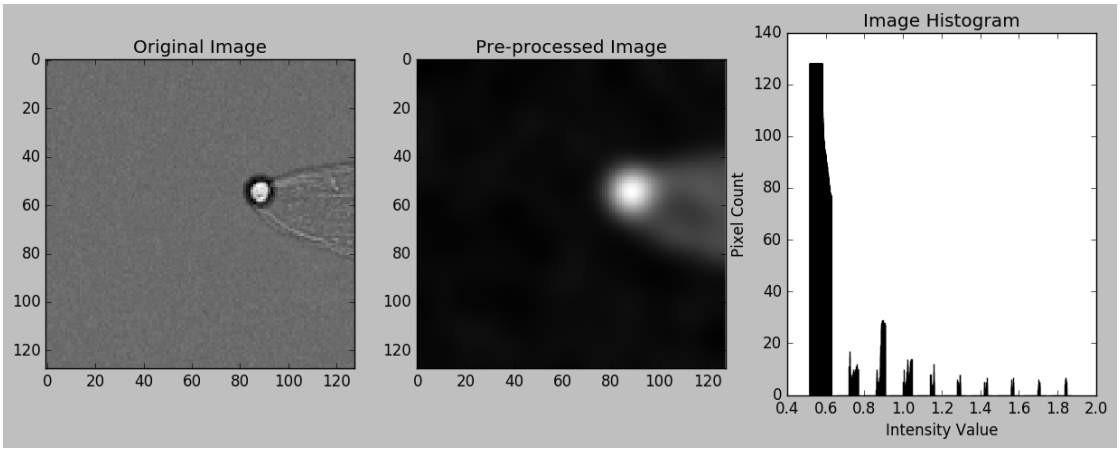

Figure 4.2: Image frame and its histogram.

filter.

4.1 Thresholding Techniques

The thresholding techniques are explained below.

4.1.1 Luminance Thresholding

In this technique, the assumption is that for the images under study, the pixels in the cell have gray-levels that are either greater or smaller than the gray level of background pixels. In such images, the cell area can be simply extracted from the background using the differences in gray-level. When the image pixel is greater than a certain threshold value, the image pixel is pronounced as a part of the cell and marked as white pixels and for image pixels smaller than the selected threshold are pronounced as background pixels and marked as black pixels. Figure 4.2 shows a typical image frame in a cell event along with its histogram.

Figure 4.2 shows two almost entirely separate peaks and intervals in the histogram. The histogram contains a strong dark region in the left-hand side of the graph that belongs to the background mainly. The bright region towards the right-hand side region of the graph belongs to the cell. This separation indicates that the cell can be extracted from the background by thresholding the image at a value near to the separation valley. The threshold is selected by calculating the mean of image intensity added with the standard deviation of image intensity. A block diagram of this method is shown in Figure 4.3.

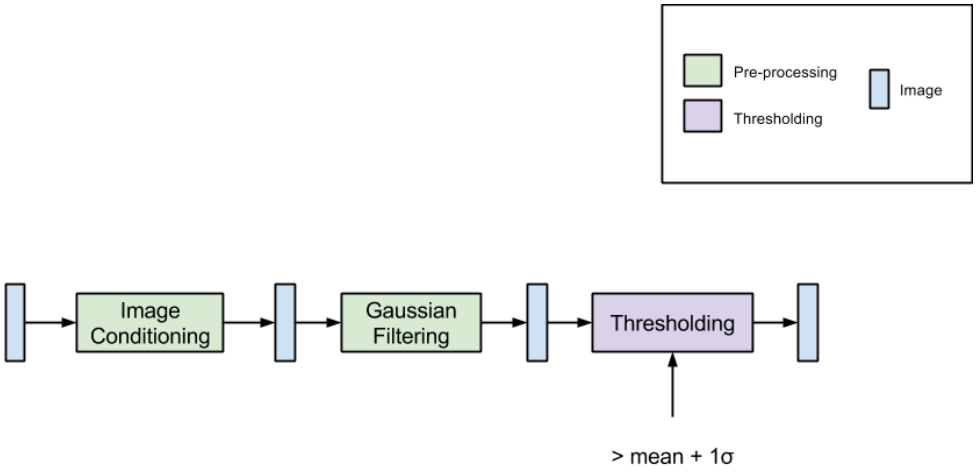

Figure 4.3: Block diagram of luminance thresholding.

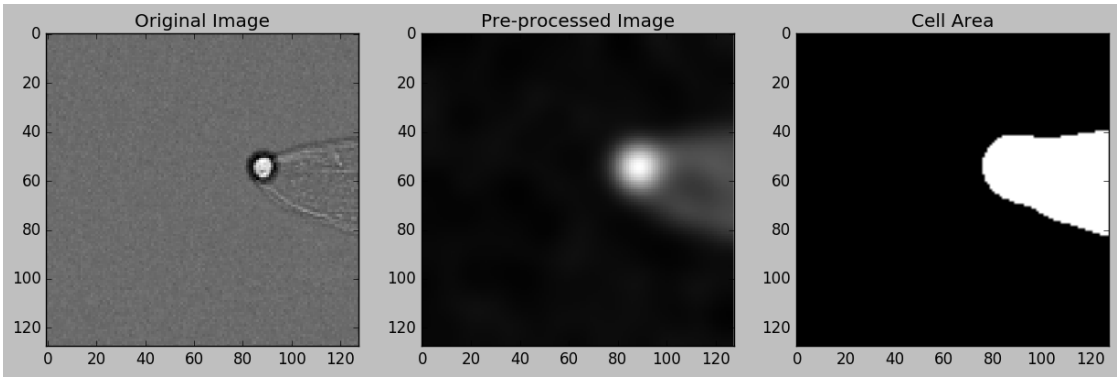

Figure 4.4: Luminance thresholding procedure.

The cell area is calculated by adding all white pixels in the thresholded image. Step-wise implementation of the above mentioned thresholding technique is shown in Figure 4.4.

4.1.2 Iterative Selection

Another technique to select an appropriate threshold is described whereby an optimum threshold is chosen automatically as a result of an iterative process [39], resulting in precise extractions of the cellular region. The threshold is calculated iteratively by an automated thresholding technique called iterative selection. In this process, an initial guess at a threshold is refined at each consecutive pass though the image [39]. This method is used to separate the cell from the background by using levels in each class repeatedly and hence improve the threshold value. The initial guess of the threshold is calculated as the mean of the image intensity. All the image

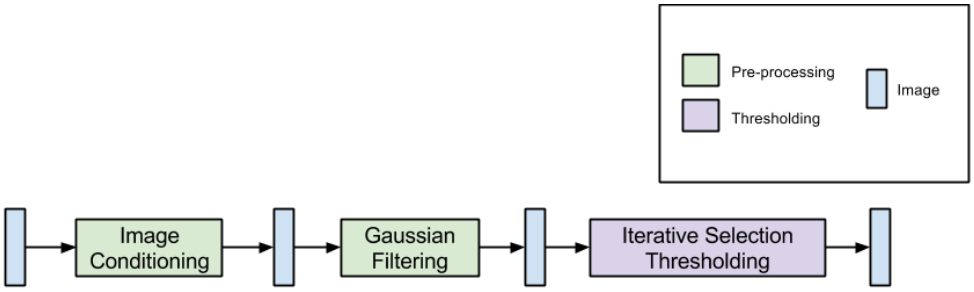

Figure 4.5: Block diagram of iterative selection.

pixels above this value are pronounced as the cell and marked as white pixels and all image pixels below this value are pronounced as the background as marked as black pixels. The mean gray-value of pixels containing the cell is calculated and called T_c and the mean gray-value of background pixels are called T_b. Now a new estimate of the threshold is computed as T_c + T_b / 2 i.e. the average of the mean levels in each pixel class. The process is repeated again using this threshold and the new and previous threshold values are compared at each iteration. The process stops when there is no change in these two threshold values through the image. The threshold value is selected as the gray-value at the last pass.

Suppose the histogram of i gray-levels in the image is h[i]. The initial estimate of the threshold is T_c. Starting with T_c, the k_{th} estimate of threshold, T_k is given by Equation 4.1.

$$\frac{\sum_{i=0}^{T_{k-1}} i.h[i]}{2\sum_{i=0}^{T_{k-1}} h[i]} + \frac{\sum_{j=T_{k-1}+1^N} j.h[j]}{2\sum_{j=T_{k-1}+1^N} h[j]} \tag{4.1}$$

When T_k = T_{k+1}, T_k is the selected threshold value. The block diagram of the above mentioned method is shown in Figure 4.5.

The cell area is calculated by adding all white pixels in the thresholded image. Step-wise implementation of the above mentioned thresholding technique is shown in Figure 4.6.

4.2 Image Segmentation Techniques

The image segmentation techniques are explained below.

4.2.1 K-means Based Clustering

K-means clustering is a divisive clustering approach, in which all data points are considered in one huge cluster and with each iteration, these clusters are recursively split to yield a

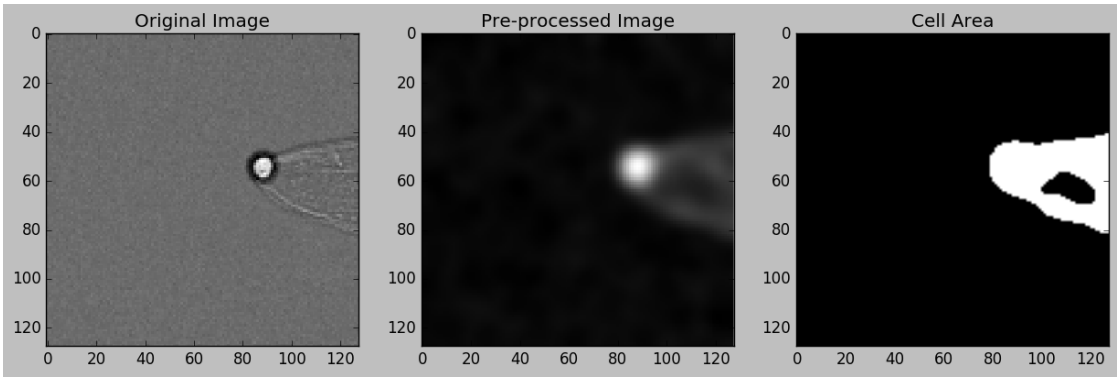

Figure 4.6: Iterative selection procedure.

good clustering [35]. Each image pixel is represented with a feature vector. The image frame consists of the location of the pixel in space is defined by the X-location, Y-location and pixel intensity. With a grayscale image with only one dimension of intensity, this gives each pixel point 3 dimensions. The feature vector consists of location x-coordinates y-coordinates, radius in polar coordinates and scaled pixel intensity values.

These feature vectors are clustered and each feature vector belongs to one cluster. The k-means algorithm tries to minimize distortion, which is defined as the sum of the squared distances between each feature vector and its dominating centroid [36]. At each iteration, the centroids are set as the mean value of data points corresponding to that cluster, and hence reduce the distortion. The change in distortion is used as a stopping criterion: when the change is lower than a threshold, the k-means algorithm is not making sufficient progress and terminates [36].

After applying k-means clustering algorithm, we reshape the labels to the shape similar to image frame. To extract the labels corresponding to the cell area, we create a binary mask from all the individual segments by looking at the average intensity of the pixels in the segment and choose a contour that has the largest perimeter. The block diagram for k-means clustering is shown in Figure 4.7.

Figure 4.8 shows k-means segmented image frames.

Varying *k*

The number of clusters, k is varied from 5-25. The original image, segmented image and binary image highlighting the cell area is shown for k = 5 in Figure 4.9, k = 10 in Figure 4.10, k = 15 in Figure 4.11 and k = 25 in Figure 4.12.

The number of pixels are added to calculate the cell area and compared against the

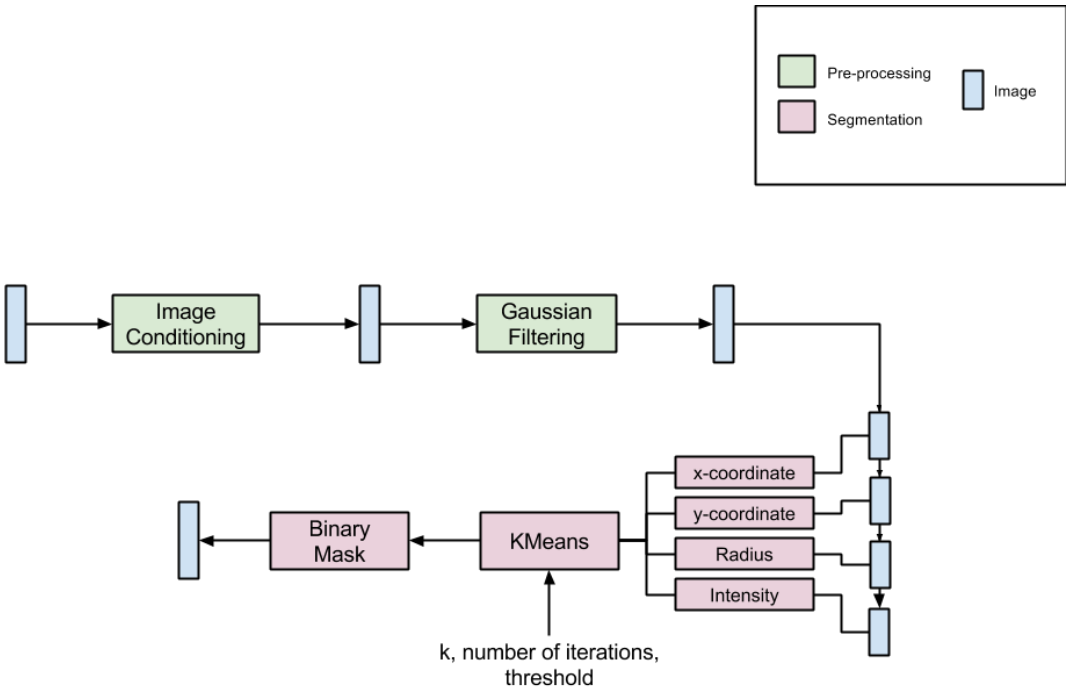

Figure 4.7: Block diagram for K-means based clustering.

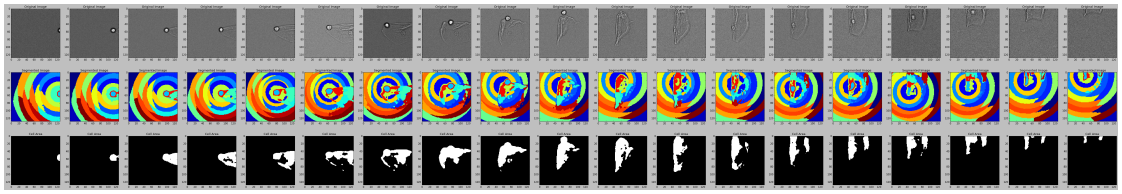

Figure 4.8: K-means based clustering with k = 20.

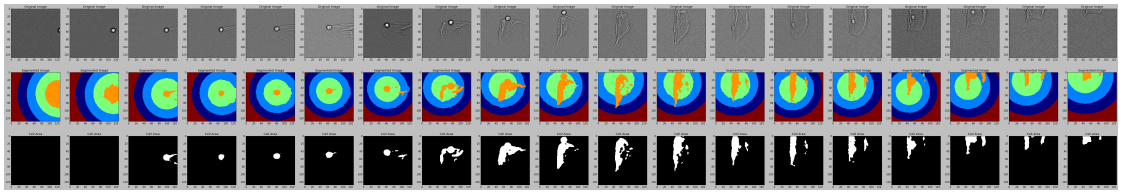

Figure 4.9: K-means based clustering with k = 5.

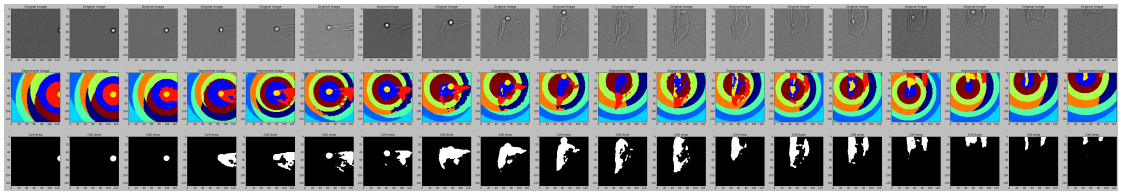

Figure 4.10: K-means based clustering with k = 10.

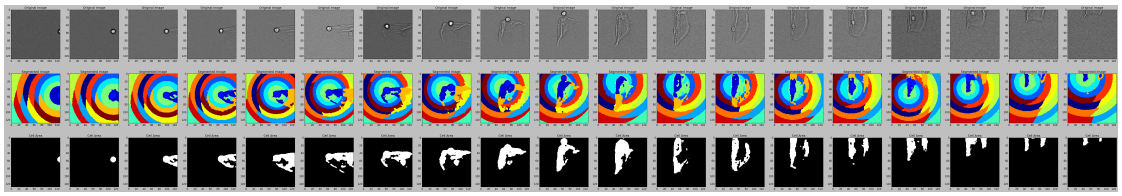

Figure 4.11: K-means based clustering with k = 15.

ground truth. The detailed procedure will be discussed in Chapter 6. K-means clustering is simple and fast in terms of complexity and processing time. But the number of clusters, maximum number of iterations and stopping threshold has to be decided beforehand and in those terms it is not an automated clustering technique.

4.3 Hysteresis Thresholding

So far, we have been looking into thresholding with respect to the intensity of each pixel in the image frame. The pixels with intensities above the selected threshold are pronounced as cell and below it is background. But, as shown from Figure 4.8, few parts of the cell, especially the surrounding cell membrane is chopped off. In case of thresholding, pixels assigned to a single class do not always form coherent regions as the spatial locations of pixels are completely disregarded. This leads to miscalculation of the cell area.

Canny, in his paper on edge detection [42] popularised the application of connectivity-based hysteresis to thresholding [41]. This method is termed as hysteresis thresholding in which

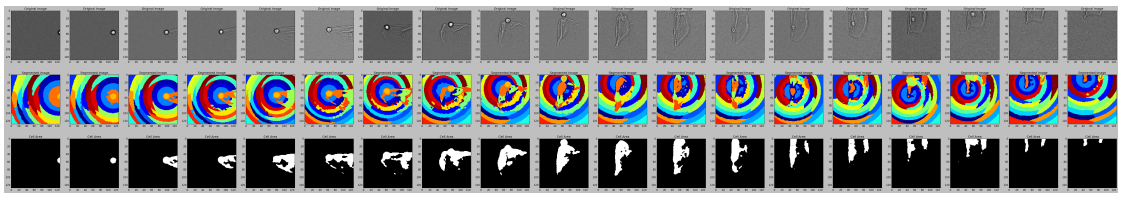

Figure 4.12: K-means based clustering with k = 25.

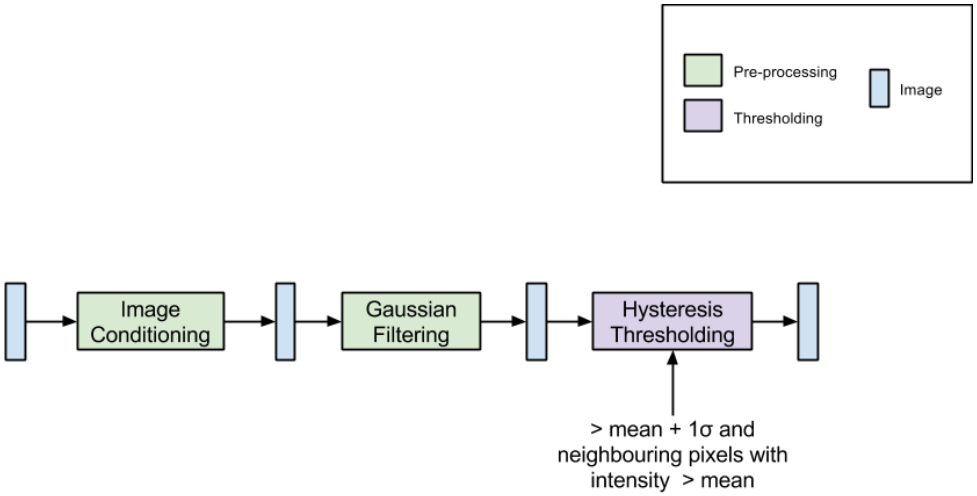

Figure 4.13: Block diagram of hysteresis thresholding.

a bi-level intensity threshold is applied, producing three regions in the thresholded image. All pixels above the high threshold are retained and set in class H, and all pixels below the low threshold are rejected and set in class L. The remaining pixels (class M) are retained only if they are adjacent to pixels in class H.

The advantage of applying hysteresis is that it incorporates spatial context into the thresholding decision [41]. In this way, the coherency of region to be detected is maintained rather than sporadic regions obtained by global thresholding.

The cell is segmented using this hysteresis thresholding method [41]. We can apply this technique of incorporating context to region thresholding as a method for eliminating the background from the cell without fragmenting the entire image frame. The processed image is thresholded at two levels, and regions in the intermediate range of intensities are rejected unless they are connected to regions generated by the higher threshold.

The block diagram of this technique is shown in Figure 4.13.

The analysis on one of the image frames from a cell event is described in this section shown in Figure 4.14.

The histogram of an image frame in cell event 23 of test dataset is shown in Figure 4.15.

4.3.1 Threshold Selection

The upper and lower threshold values are experimentally selected. The upper threshold *p1*, above which all pixels belong to the cell is set as the mean of image intensity added with

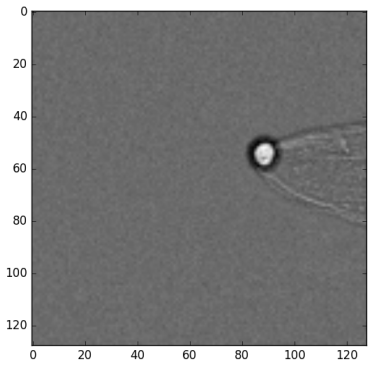

Figure 4.14: Image frame under study.

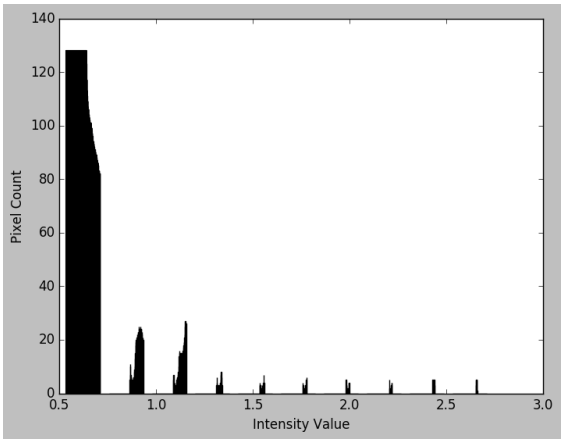

Figure 4.15: Histogram of image frame.

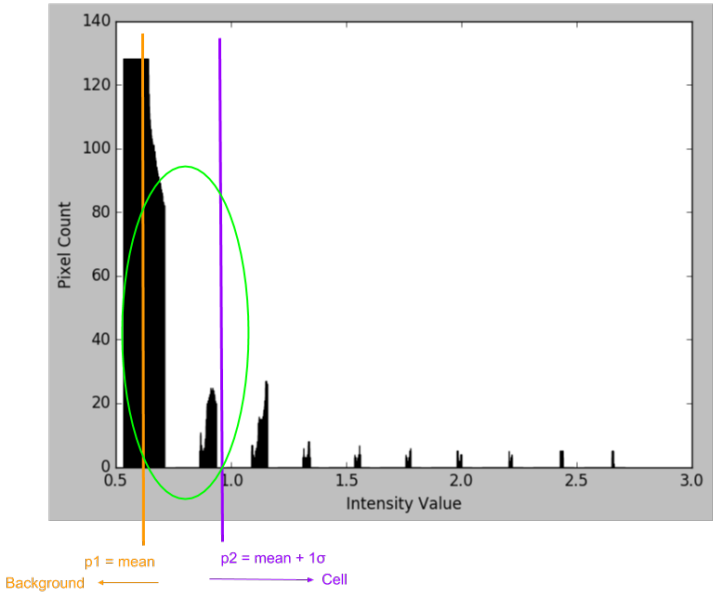

Figure 4.16: Selected threshold points for hysteresis thresholding.

the standard deviation of image intensity. The lower threshold *p2*, below which all pixels are contained in the background as the mean of image intensity. The pixels in class M are the ones in between *p1* and *p2*. The thresholding points are shown in Figure 4.16.

The detected cellular region i.e. pixels in class H are shown in Figure 4.17 and the detected background i.e. pixels in class L are shown in Figure 4.18.

4.3.2 Region Growing

To account for the surrounding cellular region, firstly the bounding box around the pixels in class H is found. Pixels in class M will be pronounced as belonging to the cell only if they lie anywhere in the surrounding region of pixels in class H. This surrounding region is controlled by a parameter, *n*, which is the window size around pixels in class H. All pixels present in pixels with class M will be pronounced as cell that are contained in the *n x n* window. The resulting image is shown in Figure 4.19.

The cell event after hysteresis thresholding is shown in Figure 4.20

The cell area is calculated as the total number of white pixels in the resultant image.

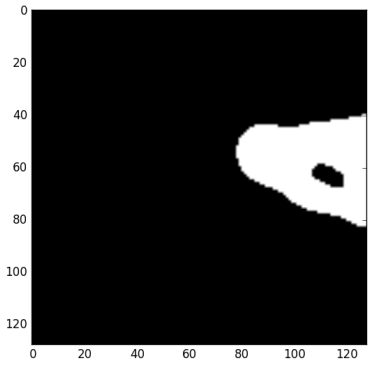

Figure 4.17: Pixels in class H (cell).

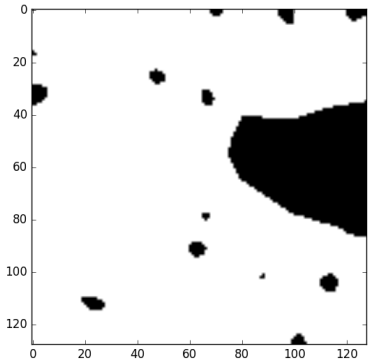

Figure 4.18: Pixels in class L (background).

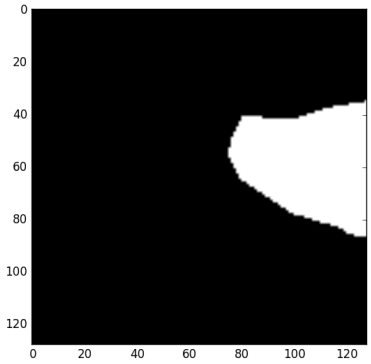

Figure 4.19: Pixels pronounced as cell after region growing.

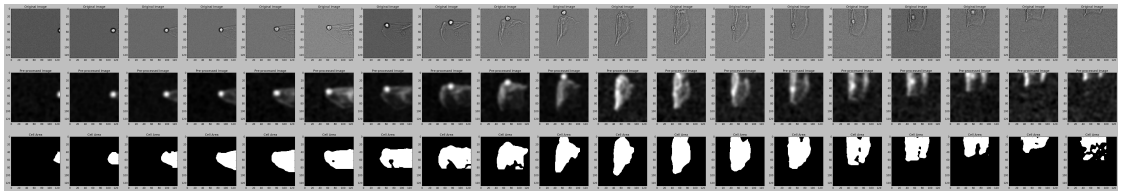

Figure 4.20: Pixels pronounced as cell in a cell event by hysteresis thresholding.

4.4 Hysteresis Thresholding with Temporal Signature

The hysteresis methodology described above attempts to combine local and global information: the two thresholds are calculated globally while the thresholding in the intermediate range uses local information [41]. To obtain a closed, coherent region for a stable selection, the temporal signature at each pixel is analyzed i.e. a sequence of values which only changes by small amounts over time. The hysteresis thresholding operation, therefore, is modified for this application to remove most of the small, high-intensity structures in close proximity to the cell nuclei [43]. To achieve this, after the bi-level thresholding step, the image with pixels of class L is added with image with pixels of class M. The temporal signature of the resultant image is processed to find the cell region.

The block diagram of this technique is shown in Figure 4.21.

The detailed procedure of calculating the cell area is explained below.

4.4.1 Threshold Selection

The histogram of an image frame in cell event 23 is shown in Figure 4.15. The upper and lower threshold values are experimentally selected. The thresholding range is stretched in this method and hence number of pixels in class M are increased which helps in attaining better control over the cell segmentation procedure. The upper threshold *p1*, above which all pixels belong to the cell is set as the mean of image intensity added with the variance of image intensity. The lower threshold *p2*, below which all pixels are contained in the background as the maximum intensity of image minus 2 times the standard deviation of image intensity. The pixels in class M are the ones in between *p1* and *p2*. The thresholding points are shown in Figure 4.22.

The detected cellular region i.e. pixels in class H are shown in Figure 4.23 and the detected background i.e. pixels in class L are shown in Figure 4.24.

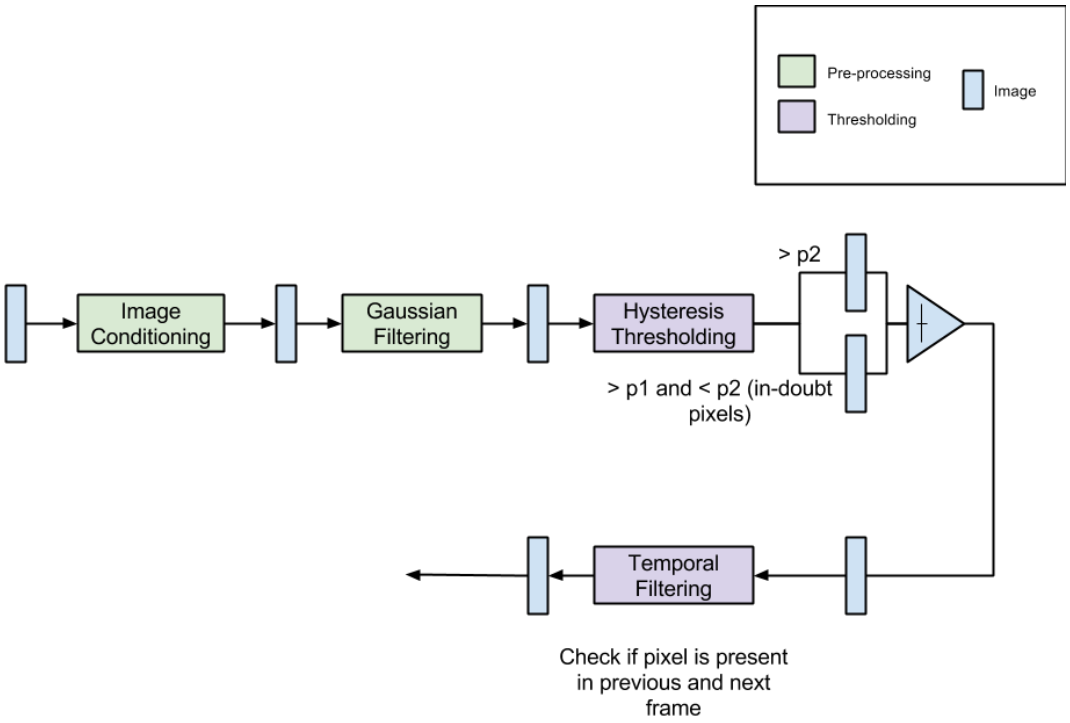

Figure 4.21: Block diagram of hysteresis thresholding with temporal signature.

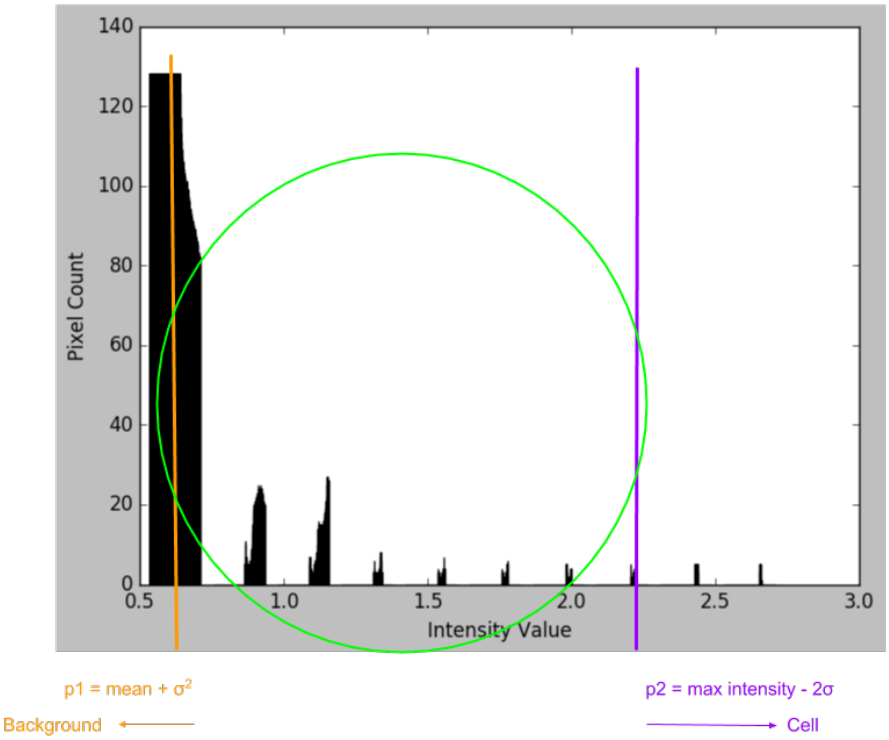

Figure 4.22: Selected threshold points for hysteresis thresholding with temporal signature.

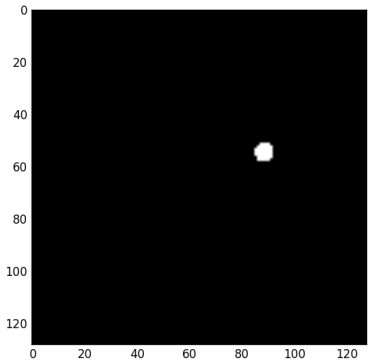

Figure 4.23: Pixels in class H (cell).

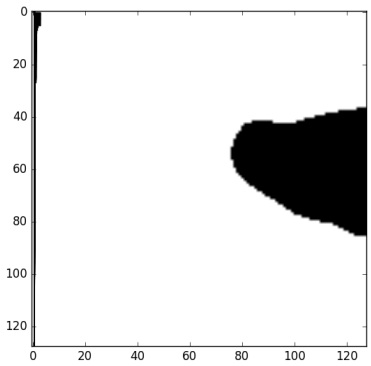

Figure 4.24: Pixels in class L (background).

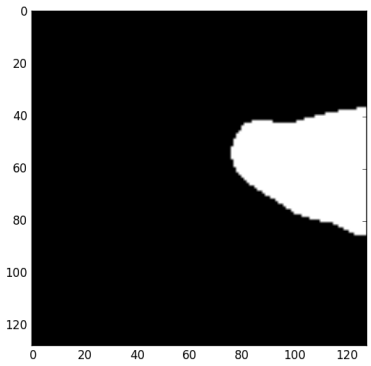

Figure 4.25: Pixels in class H and M.

4.4.2 Temporal Signature

To account for the surrounding cellular region, firstly the image of class H is added with image of class M shown in Figure 4.25.

As seen in Figure 4.26, there are some pixels around the cell and also, some background pixels are also consisted in the resultant image. At this point, temporal signature of the stream of images in a cell event can be used to determine whether the pixels in class M belong to the cell or not. The movement of the cell is steady enough to consider the position of a single pixel respective of its presence in the neighbouring image frames. The selection of such pixels is based upon whether that pixel is present in the neighbouring image frames or not.

For the first image frame, the pixel is in class M is selected only if it is present in the next two image frames. For the center image frames, the pixel in class M is selected if it is present in

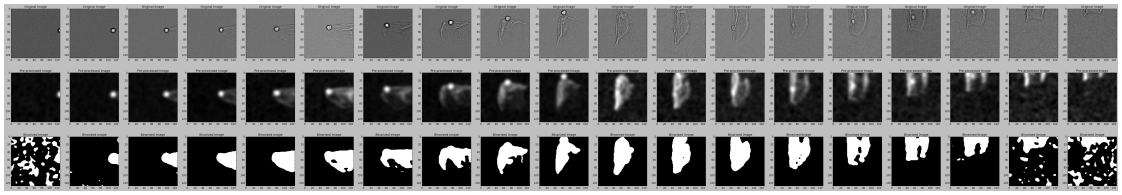

Figure 4.26: Image after bi-level thresholding.

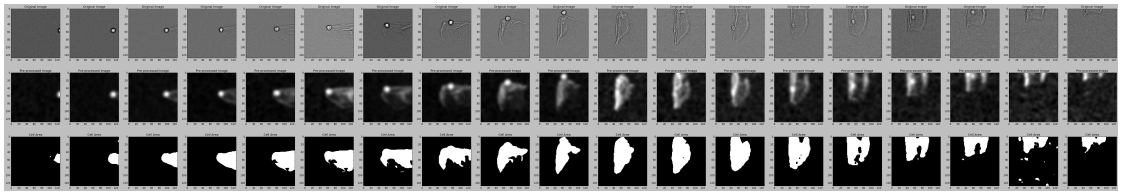

Figure 4.27: Pixels pronounced as cell in a cell event by hysteresis thresholding with temporal signature.

either the previous image frame or the next image frame. For the last image frame, the pixel is in class M is selected only if it is present in the next two image frames.The resulting image after hysteresis thresholding along with using the temporal signature is shown in Figure 4.27.

The cell area is calculated as the total number of white pixels in the resultant image.

4.5 Principal Component Analysis-based Thresholding

Another method to calculate the cell area is to apply dimensionality reduction technique like principal component analysis (PCA) [32] on the image. PCA was applied to a different test dataset with a patient run containing a set of cell worker files. On a manual inspection of these files, the images contained cells of different shapes and sizes. Cell splitting events were also visible. We developed an image analysis algorithm to calculate the cell area using dimensionality reduction techniques like PCA on the image.

An example stream of images containing a cell are shown in Figure 4.28. Each image is 40x128 pixels in resolution.

The directions of variability in the image can be visualized in a reduced image. In a 128x40 pixel image (transposed), number of samples are considered to be 128 and number of

Figure 4.28: A stream of 20 cell images.

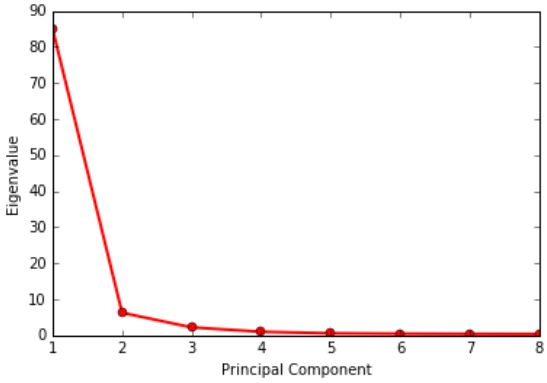

Figure 4.29: Scree plot.

features are considered across columns i.e. 40. The projection matrix resulting from PCA can be seen as a change of coordinates to a coordinate system where the coordinates are in descending order of importance [31].

To apply PCA on image data, firstly the rows are centered relative to the mean image. Then the eigenvectors corresponding to the largest eigenvalues of the covariance matrix are computed using singular value decomposition (SVD) [33]. SVD results in principal components that account for dominant directions. To find the number of dominant directions, a scree plot is used.

4.5.1 Scree Plot

A scree plot displays the eigenvalues associated with a component or factor in descending order versus the number of the component or factor [34]. This plot is useful in visually assessing the number of components that dominate most of the variability in the data. Scree plot for one of the images is shown in Figure 4.29.

From Figure 4.29, it is clear that the number of principal components for an image containing a cell is 3.

After applying dimensionality reduction, the original image is reconstructed back i.e. the data is transformed back to its original space, using directions of maximum variance in the image data. This image contains most of the information around the cellular region.

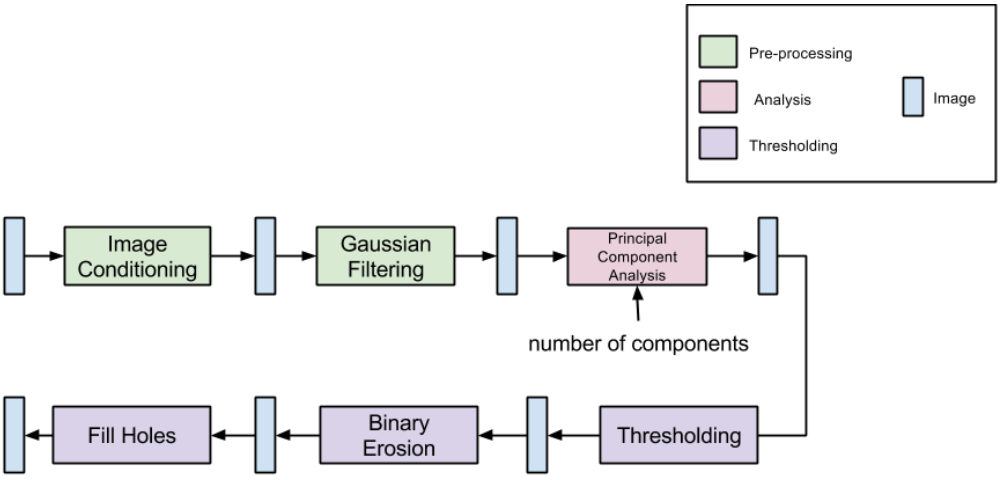

Figure 4.30: Block diagram for our method.

4.5.2 Threshold Selection

The reduced image is binarized using a thresholding constant equal to the mean of image intensity added with the standard deviation of image intensity. The noisy pixels around the cell are removed by binary erosion. Thresholding sometimes causes areas within the nucleus to fall below the threshold resulting in holes. These holes are hence filled. The total number of white pixels in the resultant binarized image is computed as cellular area.

The block diagram of this method is shown in Figure 4.30.

The results are shown in Figure 4.32 and Figure 4.34.

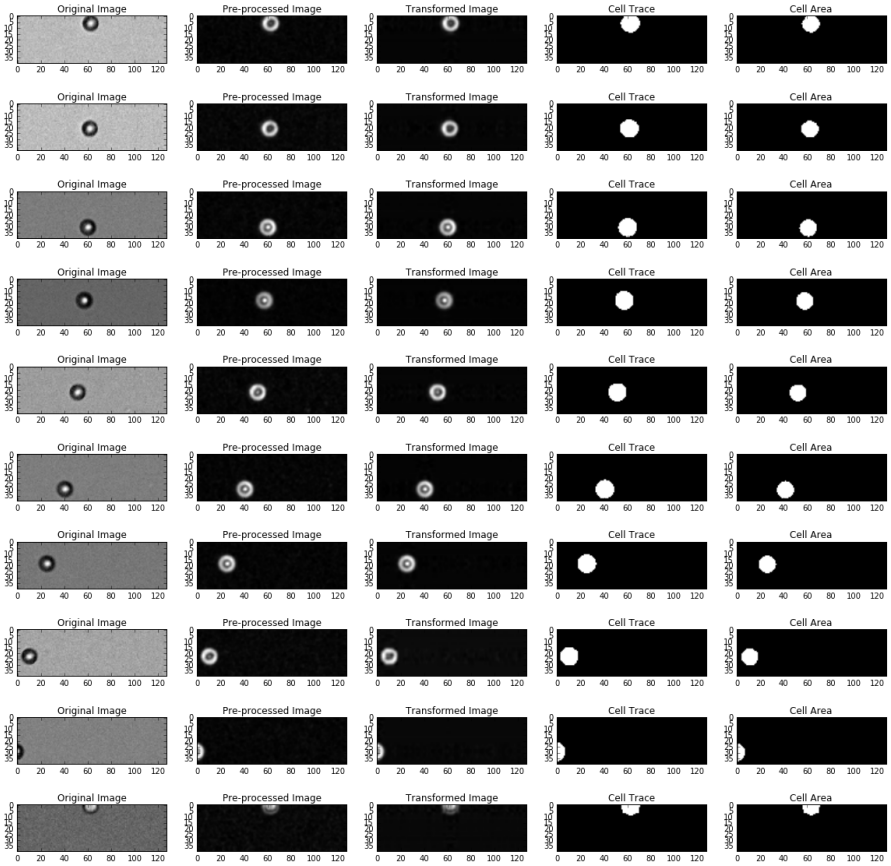

Figure 4.31: Cell area calculation using PCA - 1.

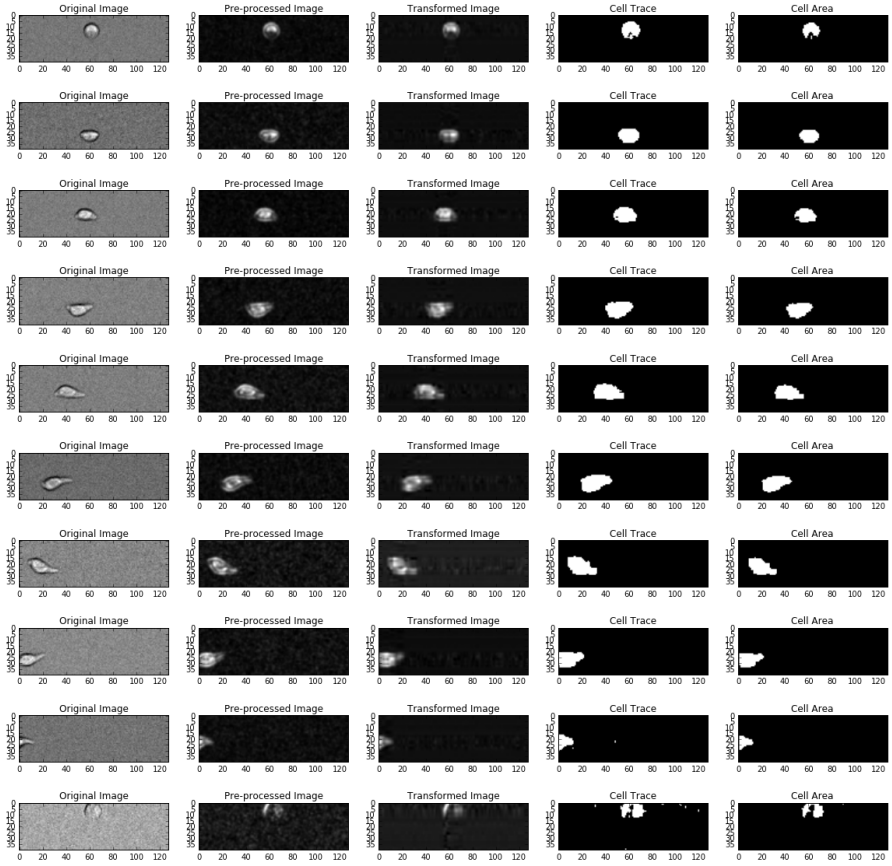

Figure 4.32: Cell area calculation using PCA - 1 (contd).

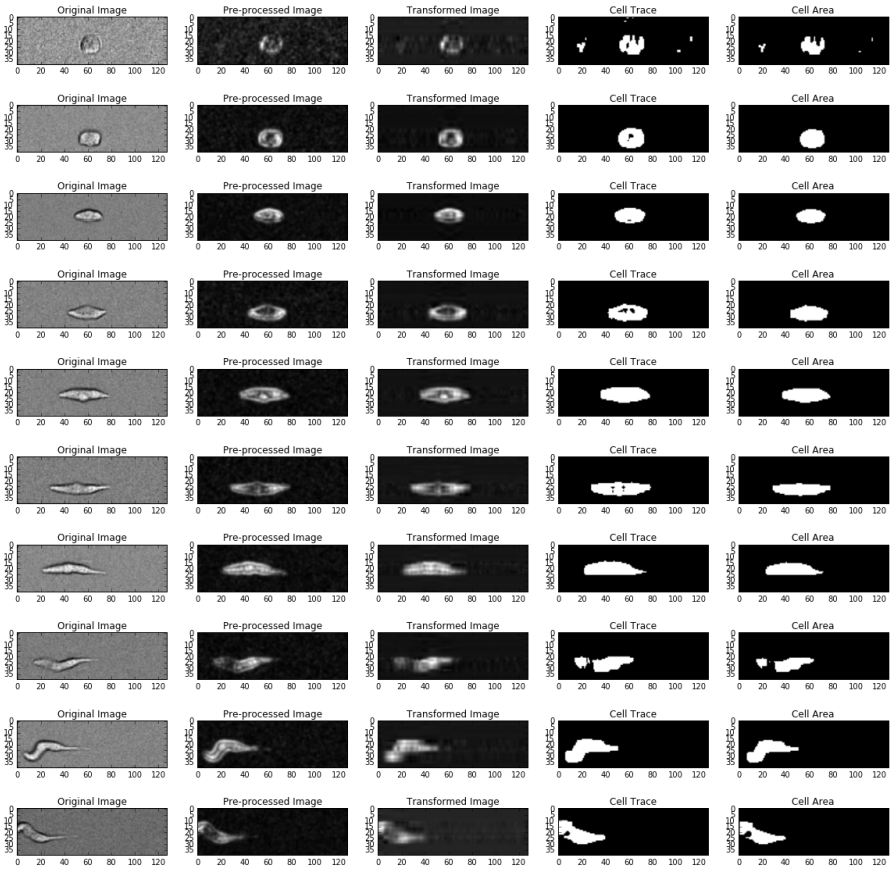

Figure 4.33: Cell area calculation using PCA - 2.

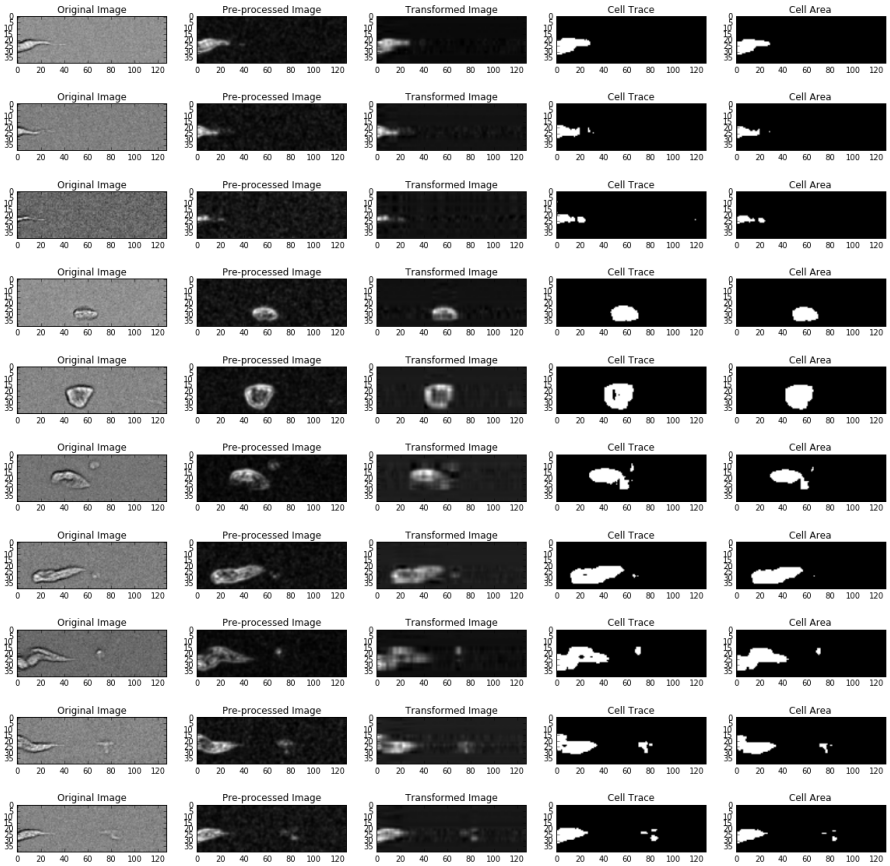

Figure 4.34: Cell area calculation using PCA - 2 (contd).

Chapter 5

Algorithms to Find Cell Nucleus Region

In the present chapter, we explore image analysis methods to describe several morphological features of the cell like nucleus size and area. Cell nuclei segmentation is a significant image analysis problem for pathology. While developing an image analysis system for detecting and describing several different subpopulations of white blood cells, a simple cell nuclei segmentation solution is explained in this chapter.

Image segmentation methods described in [44] integrate multiple metrics such as multi-scale local brightness, color, and texture cues to form a powerful globalization framework using spectral clustering. Contour detectors offer no guarantee that they will produce closed contours and hence do not necessarily provide a partition of the image into regions [44]. Due to lack of color and texture, contouring techniques implemented in [44] becomes tedious to implement on our image data. Also, due to low-resolution images, image segmentation methods like k-means or mean shift could not work well in finding the cell nuclei. These methods end up over-segmenting the image frame and cannot be useful in further processing of image data. The nuclei segmentation method proposed here, relies on the utilizing information about difference of intensities across each row and column in an image frame. This solution is designed to be robust to varying image contrast and experiment conditions.

A stream of image frames containing a cell is termed as a cell event in which each image frame contains a cell. The cell could be paired with the membrane and a nucleus. Each image frame is 128x128 pixels in resolution. A typical cell event is shown in Figure 5.1.

One of the image frames from the cell event shown in Figure 5.2, is used to describe the

Figure 5.1: A cell event under study to detect nucleus.

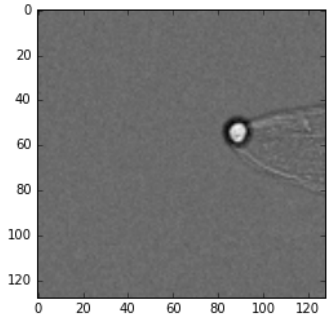

Figure 5.2: Image frame from cell event.

process of finding the cell nucleus.

The cell area is extracted by hysteresis thresholding along with temporal signature of the image frame as described in chapter 4. This image is shown in Figure 5.3.

5.1 K-means Based Segmentation

After extracting the cell area, k-means based clustering is performed on the resultant image. The procedure is similar to the one described in Chapter 4. Figure 5.4, 5.5, and 5.6 shows the result of this type of clustering on cell event 3, 10, and 23 respectively.

From the above figures, it is evident that the segmented images are not intuitive of the

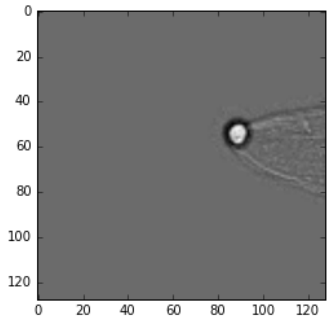

Figure 5.3: Cell area obtained by hysteresis thresholding.

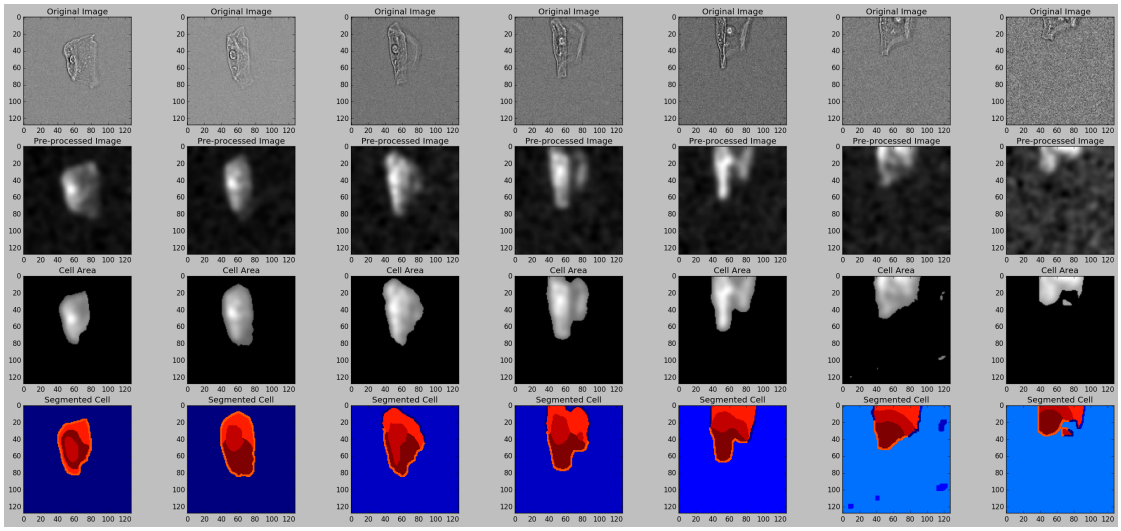

Figure 5.4: K-means segmentation for cell event 3.

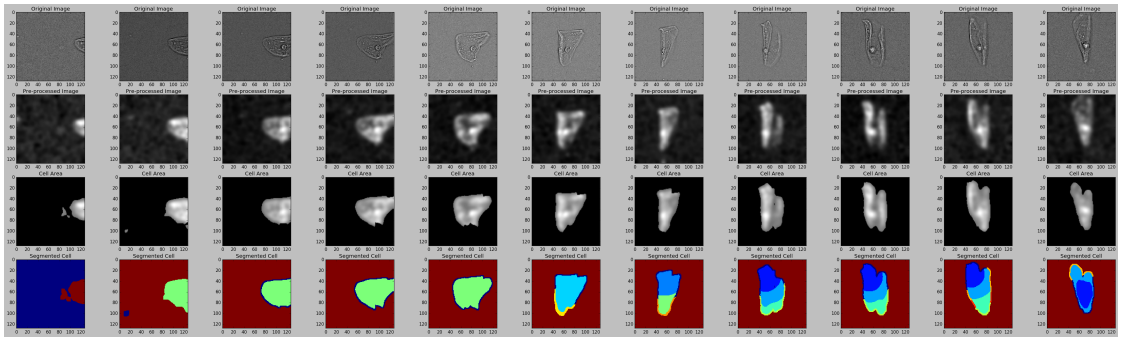

Figure 5.5: K-means segmentation for cell event 10.

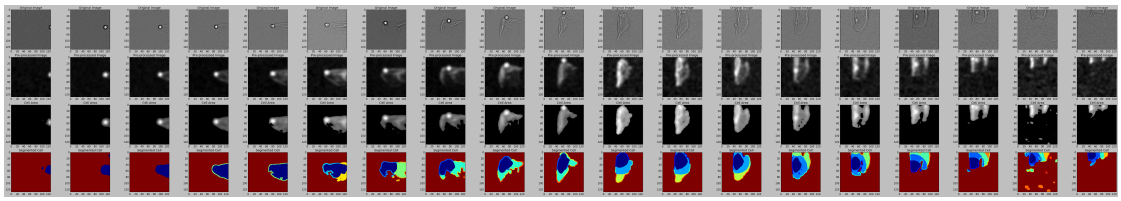

Figure 5.6: K-means segmentation for cell event 23.

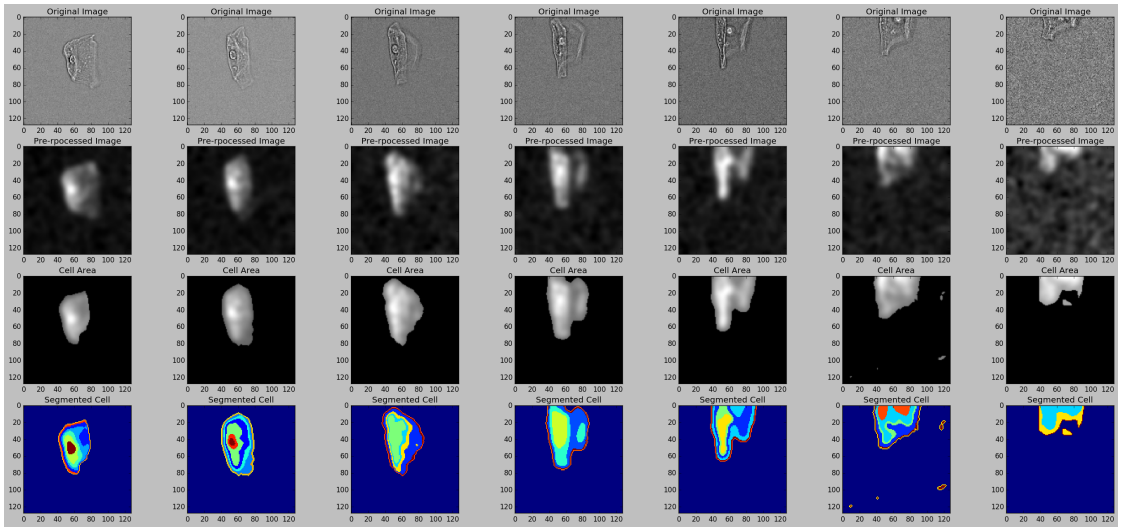

Figure 5.7: Mean shift segmentation for cell event 3.

cell nuclei region. When inspected visually, there is no cluster that belongs to the cell nuclei.

5.2 Mean Shift Segmentation

Mean shift builds upon the concept of kernel density estimation (KDE) [37]. It is assumed that intensity values of an image frame are sampled from a probability distribution. KDE is a method to estimate the underlying distribution for a set of data [37]. It works by placing a kernel on each intensity value in the image frame. Adding all of the individual kernels up generates a density function. Depending on the kernel bandwidth parameter used, the resultant density function will vary. A flat kernel is used here with a bandwidth of 7. At the end of the algorithm, significant features correspond to denser regions in the feature space [38] that represent the cellular wall and inner cell characteristics like nucleus.

Figure 5.7, 5.8, and 5.9 shows the result of this type of clustering on cell event 3, 10, and 23 respectively.

From the above figures, it is evident that the segmented image are not intuitive of the cell nuclei region. When inspected visually, there is no cluster that belongs to the cell nuclei. Hence, image segmentation methods fail to segment the images under study to extract the cell nuclei region. These methods end up over-segmenting the image frame and cannot be useful in further processing of image data.

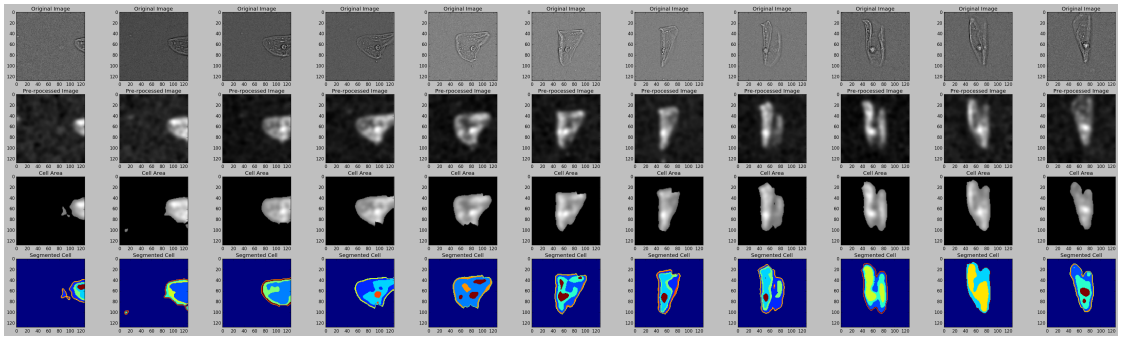

Figure 5.8: Mean shift segmentation for cell event 10.

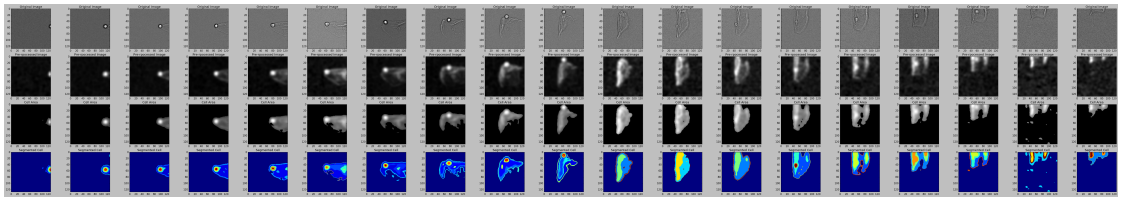

Figure 5.9: Mean shift segmentation for cell event 23.

5.3 Cumulative Intensity Difference

As seen in Figure 5.3, the likelihood of intensity fluctuation near the nucleus is more than any other part of the image. Most nuclei contain considerable detail and intensity variation as compared to the cellular region. The nucleus is contained of a bright (not necessarily circular) region surrounded by a darker ring. The transparency of the cell membrane leads to camouflaging with the background, but since we have extracted the cell area, the cell membrane is also a part of the cell. The aim of proposed solution is to find a bounding box near the cell nucleus. Figure 5.10 shows row-wise intensities in a gap of 20 rows.

It is pretty evident that intensity variation is recorded near row number 40-80. Similarly, this drill is observed across columns in the image frame. Figure 5.11 shows column-wise intensities in a gap of 20 rows.

The intensity variation is recorded near column number 80-100. To convert this eyeballing procedure into an automated process, the steps are explained in the following section.

5.3.1 Computation Procedure

The computation is motivated by the intuition of high intensity variation near the nucleus. At each row, the difference in maximum and minimum intensity is calculated. The mean

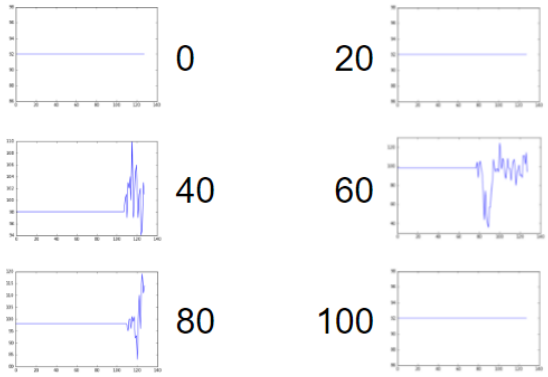

Figure 5.10: Intensities across rows.

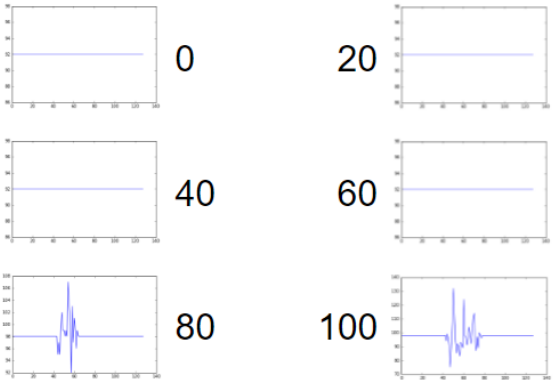

Figure 5.11: Intensities across columns.

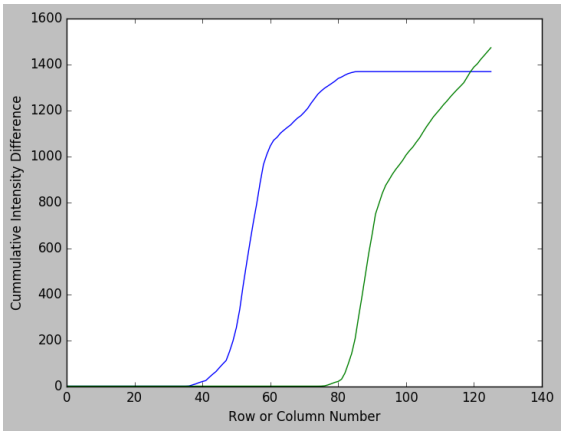

Figure 5.12: Cummulative intensity difference.

of this difference across a set of rows, say *N*, is calculated. At each data point *N*, the mean value is added to the previously calculated value. This results in a non-decreasing data vector of *image size - N* rows. Lets call this vector as *row intensity gradient, R(i)*. Equation 5.1 mathematically describes the notion of this method.

$$R(i) = \sum_{i=1}^{128-N} \frac{\sum_{k=1}^{N} Imax_k - Imin_k}{N} \tag{5.1}$$

The same procedure is repeated across columns. The plotted vectors for both rows and columns (*N*=2) is shown in Figure 5.12.

The window around the slope in Figure 5.12 corresponds to steady increase in intensity variation across rows and columns. This rectangular region is suggested to be the region surrounding the cell nuclei. The computation proceeds by finding the end-points of the bounding box, for x-coordinate end-points, the difference at each data point of *R(i)* is calculated. Lets call this vector as *row gradient*. The initial edge point, x_i is recorded as the coordinate where this difference is greater than the standard deviation of *row gradient*. The final edge point x_e is recorded as the coordinate where this difference is smaller than the standard deviation of *row gradient*. The same procedure is carried out for finding y-coordinate end-points. The partitioned cell region containing the nucleus is shown in Figure 5.13. From left, original frame, processed frame, cell-extracted frame, partitioned cell for nucleus segmentation.

Figure 5.14, Figure 5.15, and Figure 5.16 shows the nuclei region segmentation for cell event 3, 10, and 23 respectively.

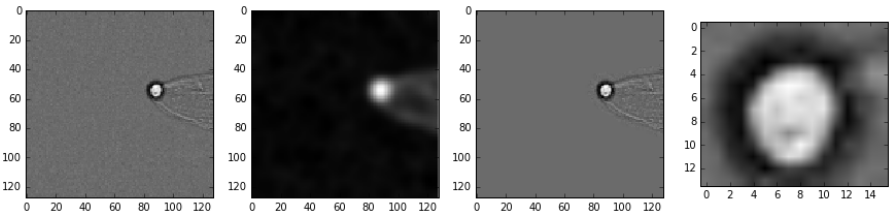

Figure 5.13: Cell partitioning.

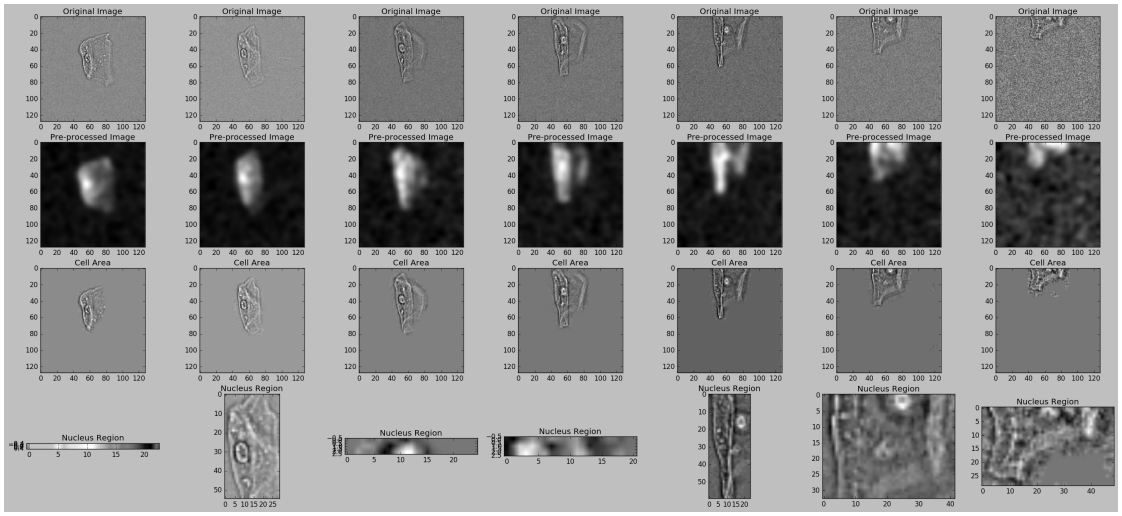

Figure 5.14: Nucleus region partitioning for cell event 3.

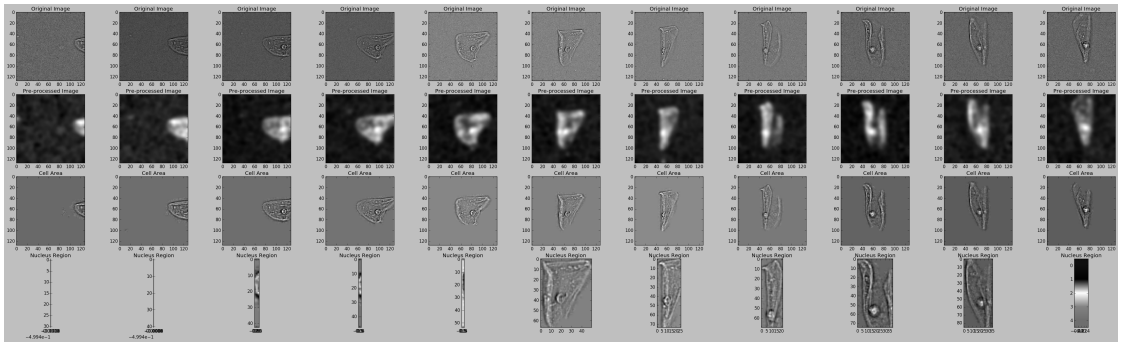

Figure 5.15: Nucleus region partitioning for cell event 10.

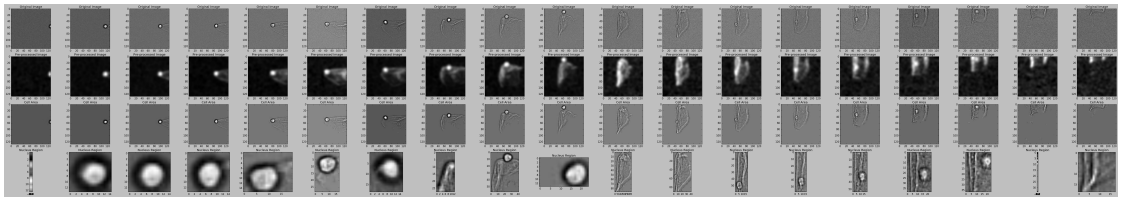

Figure 5.16: Nucleus region partitioning for cell event 23.

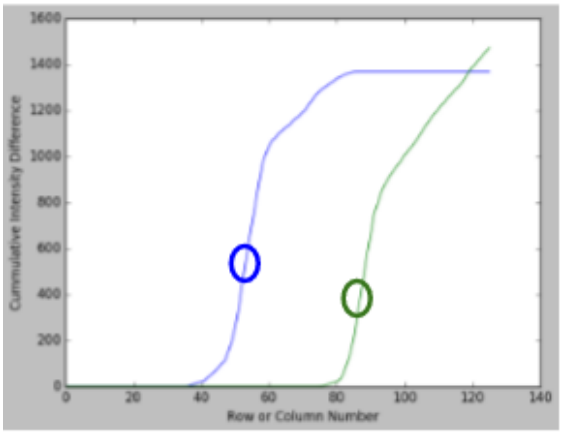

Figure 5.17: Estimated nucleus center.

5.4 Modified Cumulative Intensity Difference

Cell nuclei segmentation using the method described in the previous section sometimes fails to partition the cell around its nucleus region. This is because *row gradient* and *column gradient*, in some cases, turns out to be monotonically increasing without hitting a saturation point and hence the end point is not found. In some cases, the end point is close to the start point resulting in a thinner rectangle that clips the entire nucleus region.

In this section, we propose a modified solution to extract the nucleus region. The cell area is extracted similar to the process described in the previous section i.e. by hysteresis thresholding coupled with temporal signature information. After the cell area is extracted, *row gradient* and *column gradient* are also computed in the same way as described in the previous section. The difference lies in selecting the bounding box around the nucleus. Instead of finding the start and end points, the point at which the difference in intensities is the most, is noted. This point is shown in Figure 5.17.

This can safely be assumed to be an estimate of the center of nucleus as the intensity variation is maximum around the nucleus especially around the center. The bounding box is chosen according to the size of the window about the center estimate. We have used a square window of 20x20 to display the cell nuclei region.

Figure 5.18, Figure 5.19, and Figure 5.20 shows the nuclei region segmentation for cell event 3, 10, and 23 respectively.

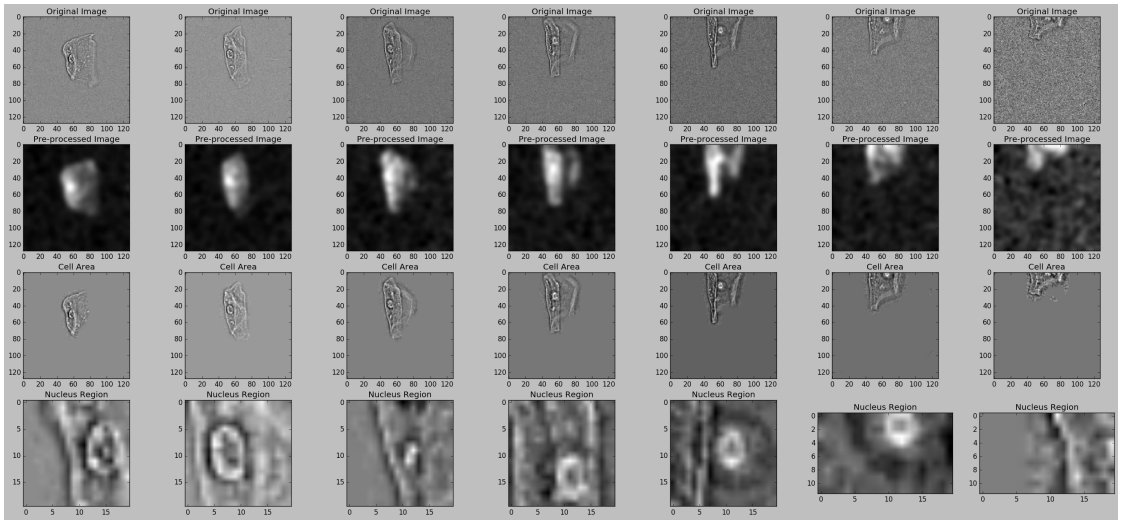

Figure 5.18: Modified nucleus region partitioning for cell event 3.

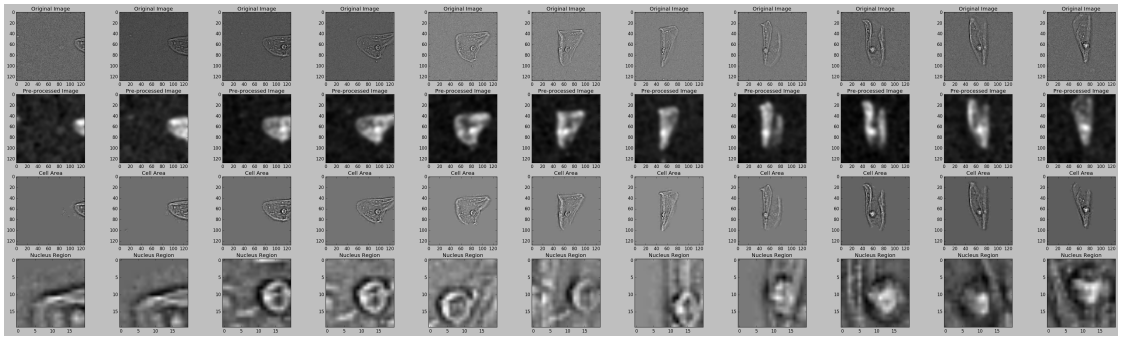

Figure 5.19: Modified nucleus region partitioning for cell event 10.

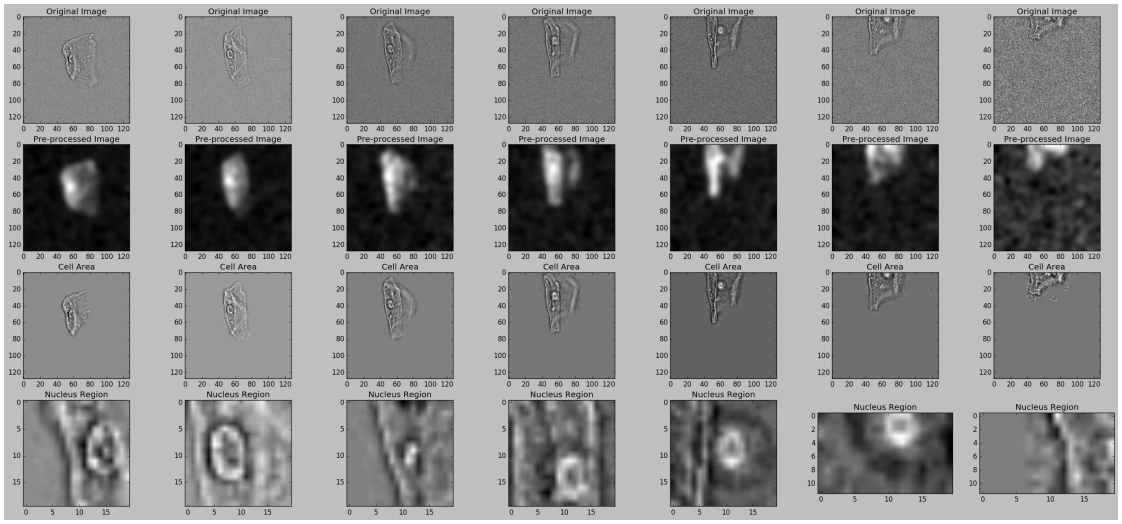

Figure 5.20: Modified nucleus region partitioning for cell event 23.

Chapter 6

Experimental Results and Future Work

In this chapter, we present our experimental results against the ground truth data. We also discuss advanced approaches to describe more details of cell populations.

Our experiments use an Intel(R) Core i5-4210U CPU at 1.70 GHz workstation for the MATLAB and Python results.

6.1 MATLAB Results

MATLAB R2015b was used to implement image processing algorithms for cell detection on the test dataset contained in provided .bin files.

6.1.1 Cell Detection - Ground Truth

We have nearly 5000 image frames in each of the four .bin files (test datasets) that may or may not contain a cell. The ground truth was collected by manually examining first 1000 frames from each of test datasets. The frame numbers that contained a cell were noted and compared against the implemented cell detection algorithms.

6.1.2 Results and Comparison

We were provided with a detection algorithm beforehand which we will call as the given method. We measured the performance and accuracy of the given and four modified cell detection methods, namely contrast stretch, histogram-based thresholding I, histogram-based thresholding II and minimum error thresholding. The four modified methods differ mainly in the way of obtaining the threshold value. Contrast stretch method uses an explicit value of threshold

Table 6.1: Table of confusion for all methods where P = 135 and N = 865 for b.bin.

Given Method		**Contrast Stretch**		**HBT I**		**HBT II**		**Min Error T**	
13	0	41	0	122	39	108	4	107	20
122	865	94	865	13	826	27	861	28	845

Table 6.2: Table of confusion for all methods where P = 170 and N = 830 for c.bin.

Given Method		**Contrast Stretch**		**HBT I**		**HBT II**		**Min Error T**	
56	0	92	1	136	30	129	15	106	4
114	830	78	829	34	800	41	815	64	826

which is varied across test datasets. In histogram-based thresholding I, the threshold value is computed by looking at the histogram of background subtracted image. In histogram-based thresholding II, the threshold value is set as the value obtained by looking at the histogram of the image detected at first. In minimum error thresholding method, the threshold value of the background subtracted image is set as the value that minimizes a criterion function for least error classification rate.

In each of the methods, after calculating the threshold value, the cell is pronounced to be present if the number of pixels in thresholded image are greater than a definite number. The frame numbers that contain a cell are noted and compared against the ground truth values.

Table of Confusion

We use the table of confusion known as the error matrix to visualize the performance of originals and 4 modified methods used for cell detection in first 1000 frames of provided test datasets. This table contains two rows and columns that reports the number of false positives, false negatives, true positives, and true negatives [46], [45]. This allows detailed analysis of the collected data and accurate comparison against the ground truth data. Tables 6.1, 6.2, 6.3 and 6.4 show the table of confusion for each of the methods. All values are in percentage.

Consider method 3 in Table 6.1 as an example. P = 135 is the number of positive instances, i.e. the number of frames in which cell is present. N = 865 is the number of negative instances, i.e. the number of frames in which cell is not present. First entry in the table is true positives (TP) = 108, i.e. the number of frames that correctly detected the presence of cell. Second entry is false negatives (FN) = 4, i.e. the number of frames that were incorrectly marked

Table 6.3: Table of confusion for all methods where P = 148 and N = 852 for d.bin.

Given Method		**Contrast Stretch**		**HBT I**		**HBT II**		**Min Error T**	
7	0	84	0	122	75	116	20	104	17
141	852	64	852	26	777	32	832	44	835

Table 6.4: Table of confusion for all methods where P = 150 and N = 850 for e.bin.

Given Method		**Contrast Stretch**		**HBT I**		**HBT II**		**Min Error T**	
11	0	100	0	119	44	110	8	98	6
139	850	50	850	31	806	40	842	52	844

as not containing a cell. Third entry is false positives (FP) = 27, i.e. the number of frames that did not contain a cell but were incorrectly marked as containing one. Last entry is true negatives (TN) = 861, i.e. number of frames that correctly detected the absence of cell.

Confusion Matrix Derivatives and Terminology

Tables 6.5, 6.6, 6.7 and 6.8 display the values in percentage of terms from derivatives of confusion matrix [47] for each of .bin files namely b, c, d, and e.bin. The definition of all the terms is as follows:

Sensitivity also known as true positive rate (TPR) is defined as

$$TPR = \frac{TP}{P} \tag{6.1}$$

Specificity also known as true negative rate (TNR) is defined as

$$TNR = \frac{TN}{N} \tag{6.2}$$

Precision also known as positive predictive value (PPV) is defined as

$$PPV = \frac{TP}{TP + FP} \tag{6.3}$$

Negative predictive value (NPV) is defined as

$$NPV = \frac{TN}{TN + FN} \tag{6.4}$$

Miss rate or false negative rate (FNR) is defined as

$$FNR = \frac{FN}{P} \tag{6.5}$$

Table 6.5: Confusion matrix derivatives for b.bin.

	Given Method	**Contrast Stretch**	**HBT I**	**HBT II**	**Min Error T**
Sensitivity	9.63	30.37	90.37	80	79.25
Specificity	100	100	95.49	99.53	97.68
Precision	100	100	75.77	96.43	84.25
NPV	87.64	90.19	98.45	96.95	96.79
Accuracy	87.8	90.6	94.8	96.9	95.2
F1 Score	17.56	46.59	82.43	87.44	81.67

Table 6.6: Confusion matrix derivatives for c.bin.

	Given Method	**Contrast Stretch**	**HBT I**	**HBT II**	**Min Error T**
Sensitivity	32.94	54.11	80	75.88	62.35
Specificity	100	99.87	96.38	98.19	99.51
Precision	100	98.92	81.92	89.58	96.36
NPV	87.92	91.4	95.92	95.21	92.8
Accuracy	88.6	92.1	93.6	94.4	93.2
F1 Score	49.55	69.96	80.95	82.16	75.71

Accuracy (ACC) is defined as

$$ACC = \frac{TP + TN}{P + N} \tag{6.6}$$

F1 score is the harmonic mean of precision and sensitivity defined as:

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{6.7}$$

All four different modifications performed better than the given method in almost all entries. There are some cases for which the modified methods worked poorly, i.e. in terms of specificity. The given method has less number of false discoveries but has missed a significant number of frames that contained a cell. Contrast stretch method works poorly because of constant threshold value applied across files with varying illumination levels.

Consider Table 6.5 as an example. The sensitivity of given method is as low as 9.63% which means that it resulted in large number of false negatives. Sensitivity of contrast stretch

Table 6.7: Confusion matrix derivatives for d.bin.

	Given Method	**Contrast Stretch**	**HBT I**	**HBT II**	**Min Error T**
Sensitivity	4.72	56.75	82.43	78.37	70.27
Specificity	100	100	91.19	97.65	98.01
Precision	100	100	61.92	85.29	85.95
NPV	85.8	93.01	96.76	96.29	94.99
Accuracy	85.9	93.6	89.9	94.8	93.9
F1 Score	9.03	72.41	70.72	81.69	77.32

Table 6.8: Confusion matrix derivatives for e.bin.

	Given Method	**Contrast Stretch**	**HBT I**	**HBT II**	**Min Error T**
Sensitivity	7.33	66.67	79.33	73.33	65.33
Specificity	100	100	94.82	99.05	99.29
Precision	100	100	73	93.22	94.23
NPV	85.94	94.44	96.29	95.46	94.19
Accuracy	86.1	95	92.5	95.2	94.2
F1 Score	13.67	80	76.03	82.08	77.16

Table 6.9: Time taken per frame.

	Contrast Stretch	**HBT I**	**HBT II**	**Min Error T**
Time taken (in seconds)	0.0033	0.26	0.15	0.15

Table 6.10: Time taken per cell event.

	Contrast Stretch	**HBT I**	**HBT II**	**Min Error T**
Time taken (in seconds)	0.063	4.44	1.31	1.14

method is better than given one but much less than the sensitivity of other three methods. This is because even after adjusted the contrast of the image, non-adaptiveness of threshold value fails to correctly detect the presence of cell in most cases. Overall, histogram-based thresholding works better than the rest of the methods. F1-score of this method is larger than that of all other methods.

The best performance by histogram-based thresholding 2 method indirectly shows the validity of the modification and the importance of the threshold value. In every table, we see a significant performance difference between the original and modified methods.

Processing Time

Table 6.9 shows the time taken in seconds per frame by each of the modified methods to detect the presence of a cell. The processing time is computed by using tic-toc feature in MATLAB R2015b. tic starts a stopwatch timer to measure performance by recording the internal time at execution of the tic command. It displays the elapsed time with the toc function [48]. Contrast stretch method is least expensive while histogram-based thresholding I method takes maximum time to process. This is because it computes the histogram of each image and decides the threshold value from the computed histogram. Histogram-based thresholding II method works faster than histogram-based thresholding I method as now the histogram for only one image is computed.

Table 6.10 shows the time taken in seconds per 1000 frames by each of the modified methods to detect the presence of a cell.

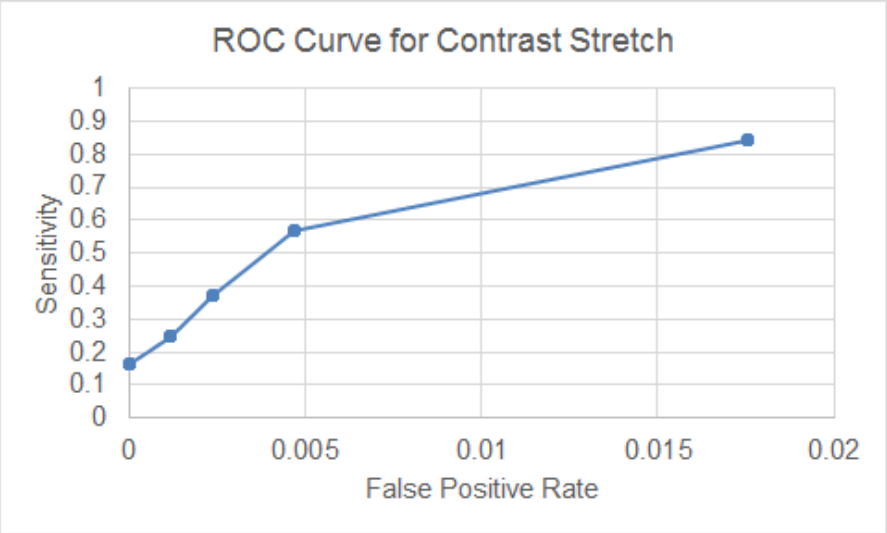

Figure 6.1: ROC curve for contrast stretch method.

6.1.3 ROC Curves

Contrast Stretch Method

The ROC curve was generated for contrast stretch method shown in Figure 6.1. Here, we adjust the original image first, and then apply thresholding on it to find the presence of a cell. The varying parameter to assess the sensitivity and false positive rate was set as the threshold that is used to binarize the adjusted image.

Histogram-based Thresholding I

The ROC curve was generated for histogram-based thresholding I method shown in Figure 6.2. Here, we set the threshold on the basis of studying histogram of each of background subtracted images. The varying parameter to assess the sensitivity and false positive rate was set as the threshold that is calculated from the histogram, finally used to binarize the background subtracted image.

Histogram-based Thresholding II

The ROC curve was generated for histogram-based thresholding II method shown in Figure 6.3. Here, we set the threshold on the basis of studying histogram of only one of background subtracted images. The varying parameter to assess the sensitivity and false positive

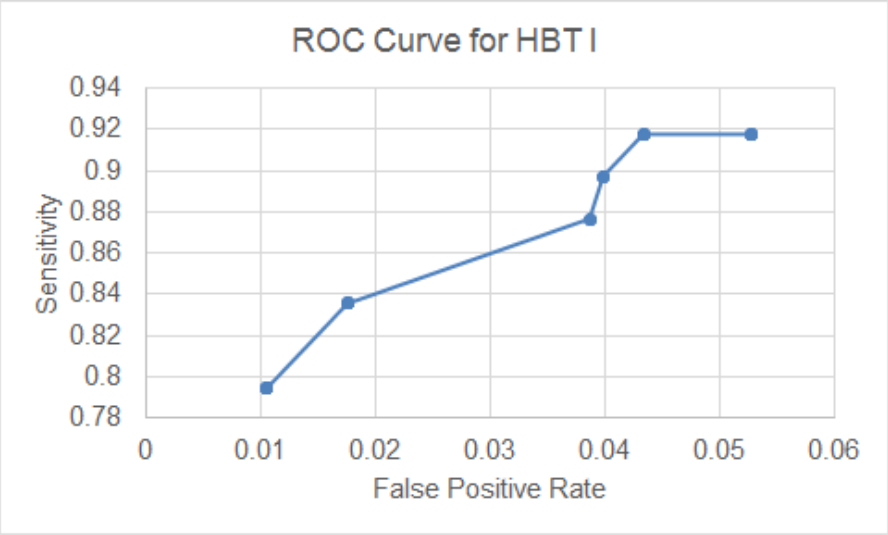

Figure 6.2: ROC curve for HBT I method.

rate was set as the threshold that is calculated from the histogram, finally used to binarize the background subtracted image.

To sum up, all three curves are presented in one graph in Figure 6.4.

Hence, we can safely conclude that histogram-based thresholding II method works the best of all the methods assessed for cell detection.

6.2 Python Results

Python 2.7 was used to implement image processing algorithms to compute cell morphology features like cell area and nucleus isolation.

6.2.1 Cell Area - Ground Truth

We have 27 cell events that contain a cell. The ground truth was collected by using *ImageJ* to manually examine all image frames in the given cell events. The cell area was noted by measuring the number of pixels contained in the hand-drawn region around the cell.

6.2.2 Results and Comparison

We measured the performance of five different methods to calculate the cell area, namely luminance thresholding, iterative selection, K-means based thresholding, hysteresis thresholding, hysteresis thresholding using temporal signature. These methods differ mainly in the way

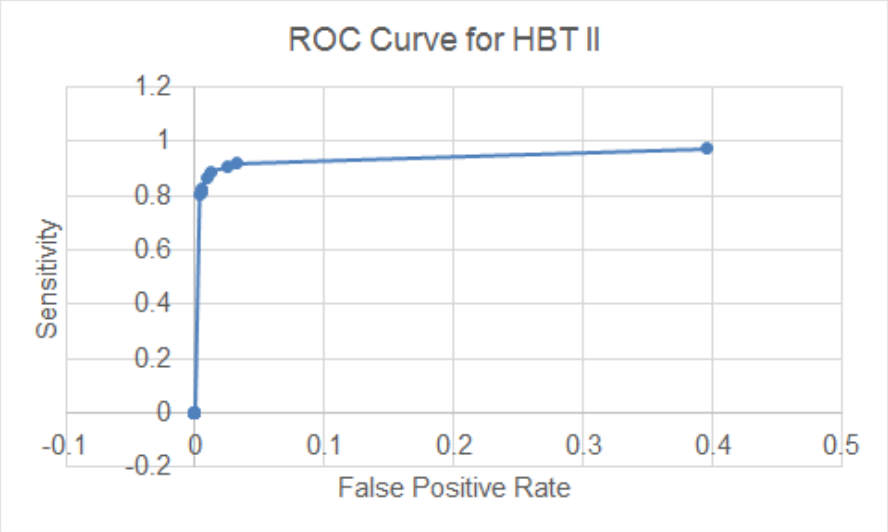

Figure 6.3: ROC curve for HBT II method.

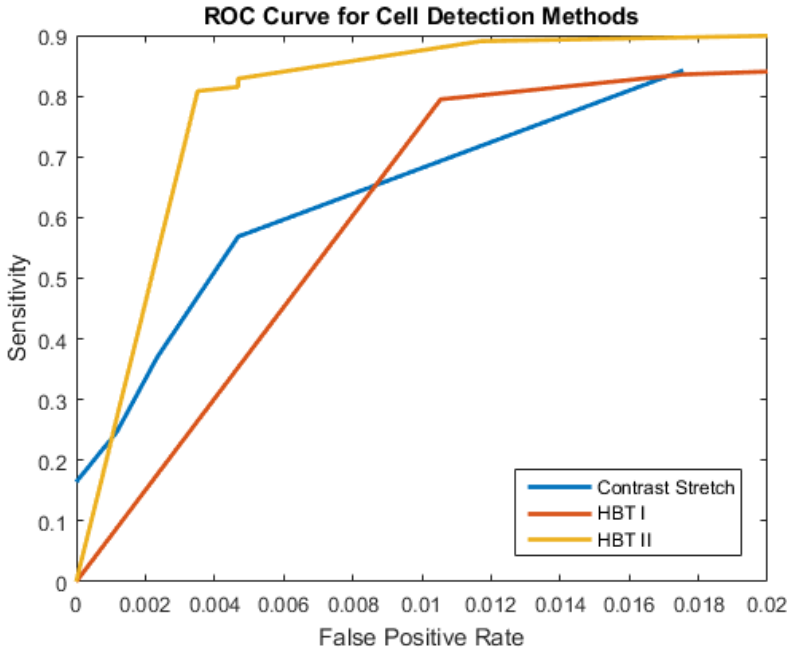

Figure 6.4: ROC curve for all cell detection methods.

of thresholding the pre-processed image for background subtraction. Luminance thresholding binarizes the pre-processed image at a pixel value of mean of image intensity added with standard deviation of image intensity.

Iterative selection recursively divides the image in foreground and background areas and finds an equilibrium point at which average of $\mu_{background}$ and $\mu_{foreground}$ at successive iterations are equal. In K-means based thresholding, the image is segmented using k-means clustering algorithm and the cluster with maximum perimeter is chosen as the cell trace. Hysteresis thresholding uses multiple thresholding points to divide the image in foreground and background class and the pixels possessing the intensity value in neither of the regions are chosen to be a part of cell if they are in close proximity of foreground pixels. Hysteresis thresholding uses temporal signature of the image frame under study to decide for the pixels that belong to neither foreground or background class.

In each of the methods, total number of pixels in the resultant binarized image is defined as the cell area. The total number of pixels are noted and compared against the ground truth values.

Figure 6.5, 6.6 and 6.7 shows a visual comparison of the computed cell area for events 1, 15, 23 respectively. The continuous line is the ground truth as labeled in all figures.

Consider Figure 6.7 as an example. First three methods work well only in cases where the cell boundary is profound. In case where the cell membrane camouflages with the background, binarized image loses that content and hence the cell area is much less than the actual number. Hysteresis thresholding works well as it accounts for proximity of pixels around the more profound cell region. Hysteresis thresholding with temporal signature works the best in almost all cell events as it also accounts for the presence of pixel in neither of the classes across its neighboring images.

Figure 6.8 shows the plot of error in number of pixels for all 320 images that were tested against this method.

6.2.3 PCA-based Computation

We were provided with a patient run containing a set of cell worker files. PCA was applied to this new set of images. Each image is 40x128 pixels in resolution. The ground truth was collected by using *ImageJ* to manually examine all image frames in the given cell events. The cell area was noted by measuring the number of pixels contained in the hand-drawn region around the cell.

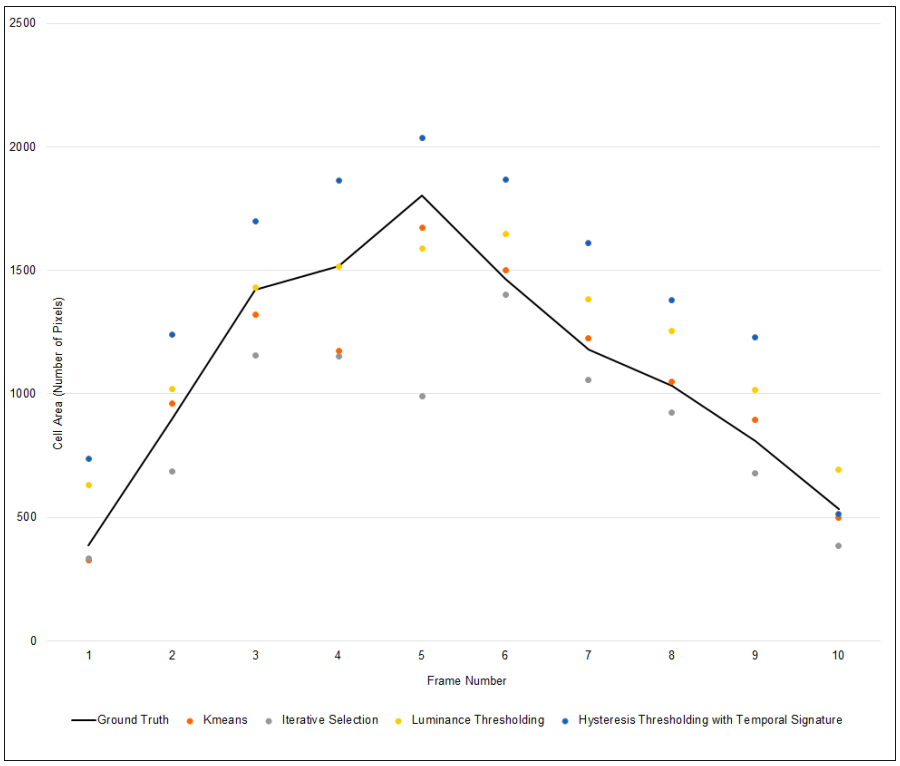

Figure 6.5: Comparison of calculated cell area for cell event 1.

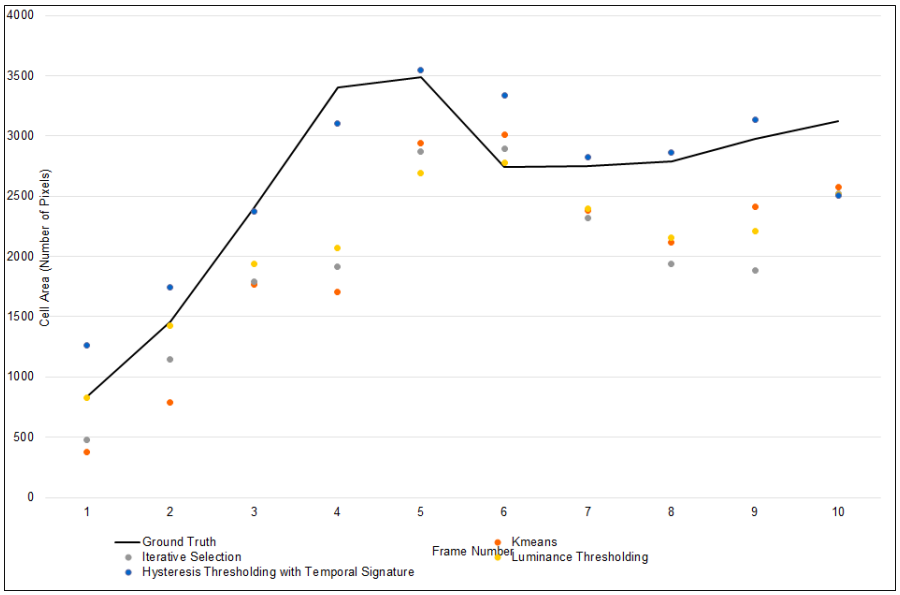

Figure 6.6: Comparison of calculated cell area for cell event 15.

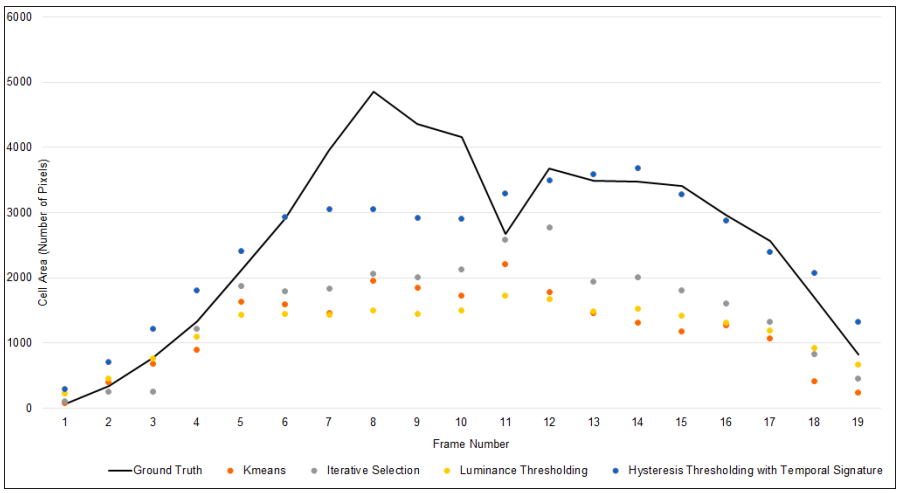

Figure 6.7: Comparison of calculated cell area for cell event 23.

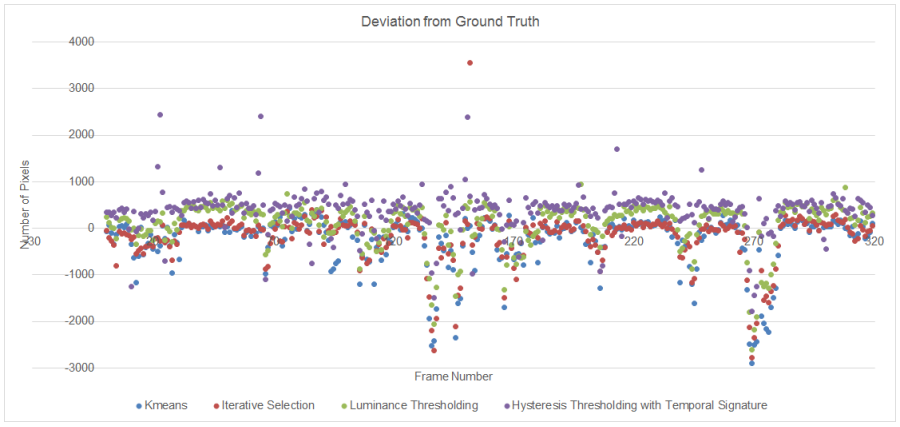

Figure 6.8: Deviation from ground truth.

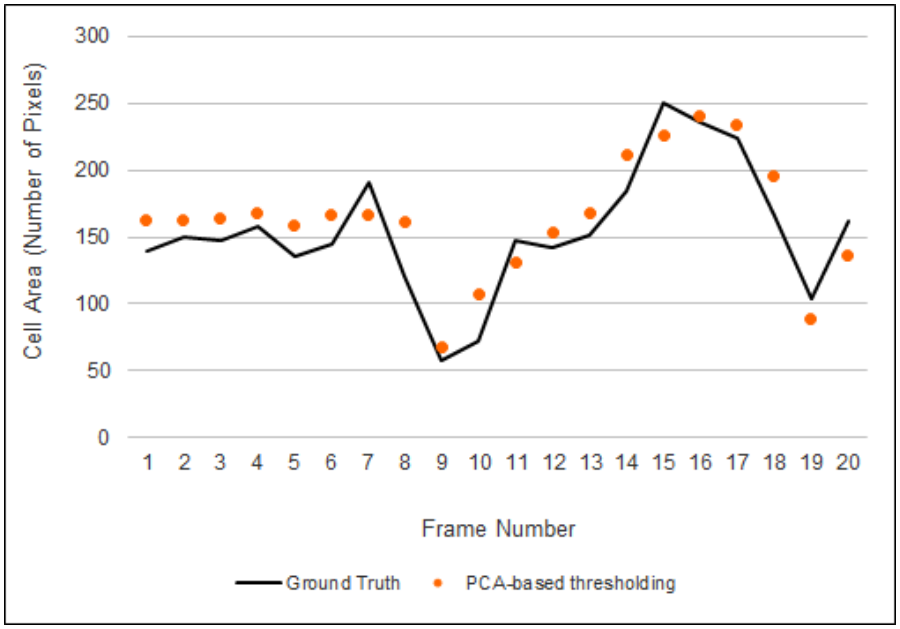

Figure 6.9: Comparison of calculated cell area for first 20 cell events.

Results and Comparison

We measured the performance of this method against the ground truth data on the basis of computed cell area.

Figure 6.9 and 6.10 shows a visual comparison of the computed cell area for a total of 40 cell events. The continuous line is the ground truth as labeled in all figures.

Figure 6.11 shows the plot of error in number of pixels for 40 images that were tested against this method.

6.3 Conclusion

The methods discussed in this thesis helps in a judicious and informed selection of threshold. This value of threshold is effective in transforming a gray-scale image into a binary image that contains only those image points that are associated with the cellular area or the nucleus area in nucleus detection case. The image pixels corresponding to cellular region assume value one and the background pixels assume the value zero. To detect the presence of cell, the histogram-based thresholding method gave exceptional results. To account for the cellular in a cellular specimen of bladder epithelial cells, hysteresis thresholding using temporal signature proved to be the best bet. Cell nucleus was successfully isolated by the cumulative intensity

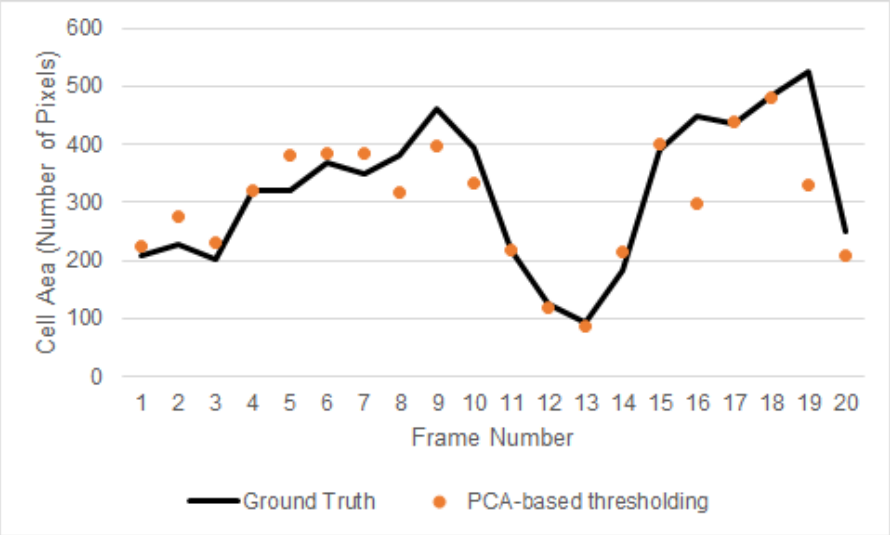

Figure 6.10: Comparison of calculated cell area for another 20 cell events.

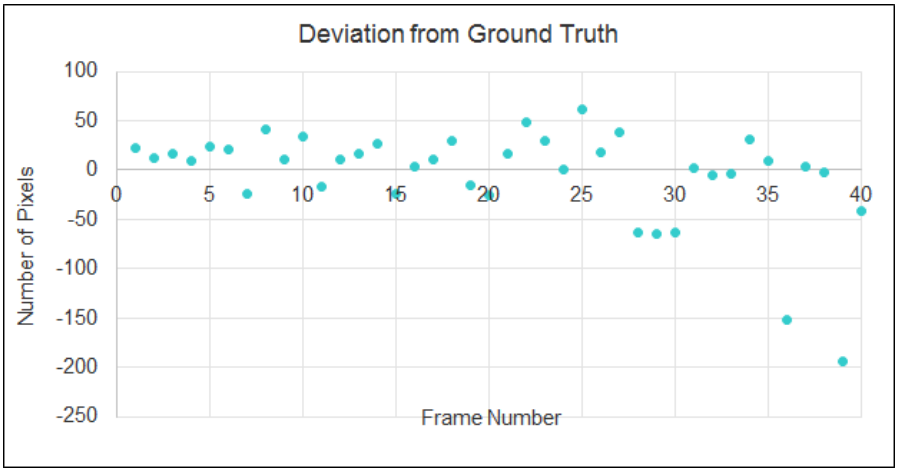

Figure 6.11: PCA - Deviation from ground truth.

difference method that exploited the variation of image intensities. In general, image intensity statistics help to extract meaningful information from small-sized microscopic images containing low image contrast. The thresholding methods described in this thesis seem simplistic, but are indeed important and fundamental to proceed towards faster image analysis. These methods help in quick construction of robust features for further cell classification.

6.4 Future Work

We aim to characterize the nucleus according to the isolated region obtained by our methods. Our ongoing studies also include implementing image analysis algorithms on raw video files of different cell subpopulations. The deformation experienced by cells in the current platform is different from the previous implementations and hence will require alternative segmentation approaches. With this thesis, we examined different image analysis technique on different image sets. However, we now aim to develop a consistent image analysis module that describes cell morphology across varied image sets. After detecting the presence of a cell and computing the cell area, we wish to extract information about cell shapes and feed this information to a classifier. Supervised machine learning algorithms would be useful to sort the cells based on their shape that are subject to deformation. Another important area that needs to be explored is the detection of cell splitting event at the device junction. The image processor can also be used to track the lineage and cell cycle phase times of each cell in a large group, and differentiate between healthy and unhealthy cells.

In addition, we aim to develop fast image processing algorithms that can be translated to lower level optimized code such as C to decrease the processing time. This would help in building a robust infrastructure for developing modularized hardware solutions.

I am grateful to Dajung, Ryan and Alexandria for co-authoring the following paper with me. Lee, Dajung; Mehta, Nirja; Shearer, Alexandria; Kastner, Ryan “A Hardware Accelerated System for High Throughput Cellular Image Analysis,” Journal of Parallel and Distributed Computing, submitted.

Bibliography

[1] Price, Jeffrey H., and David Gough. "Operator independent image cytometer." U.S. Patent 5,548,661, issued August 20, 1996.

[2] Bull, B. S., M. A. Schneiderman, and George Brecher. "Platelet counts with the Coulter counter." American journal of clinical pathology 44, no. 6 (1965): 678-688.

[3] Ortyn, William E., David A. Basiji, Philip Morrissey, Thaddeus George, Brian Hall, Cathleen Zimmerman, and David Perry. "Blood and cell analysis using an imaging flow cytometer." U.S. Patent 7,925,069, issued April 12, 2011.

[4] Tse, Henry Tat Kwong, Pingfan Meng, Daniel R. Gossett, Ali Irturk, Ryan Kastner, and Dino Di Carlo. "Strategies for implementing hardware-assisted high-throughput cellular image analysis." Journal of the Association for Laboratory Automation 16, no. 6 (2011): 422-430.

[5] Carpenter, Anne E., Thouis R. Jones, Michael R. Lamprecht, Colin Clarke, In H. Kang, Ola Friman, David A. Guertin, Joo Han Chang, Robert A Lindquist, Jason Moffat, Polina Golland and David M Sabatini. "CellProfiler: image analysis software for identifying and quantifying cell phenotypes." Genome biology 7, no. 10 (2006): R100.

[6] Blasi, Thomas, Holger Hennig, Huw D. Summers, Fabian J. Theis, Joana Cerveira, James O. Patterson, Derek Davies, Andrew Filby, Anne E. Carpenter, and Paul Rees. "Label-free cell cycle analysis for high-throughput imaging flow cytometry." Nature communications 7 (2016).

[7] Gossett, Daniel R., T. K. Henry, Serena A. Lee, Yong Ying, Anne G. Lindgren, Otto O. Yang, Jianyu Rao, Amander T. Clark, and Dino Di Carlo. "Hydrodynamic stretching of single cells for large population mechanical phenotyping." Proceedings of the National Academy of Sciences 109, no. 20 (2012): 7630-7635.

[8] Darling, Eric M., and Dino Di Carlo. "High-throughput assessment of cellular mechanical properties." Annual review of biomedical engineering 17 (2015): 35-62.

[9] Lee, Dajung, Pingfan Meng, Matthew Jacobsen, Henry Tse, Dino Di Carlo, and Ryan Kastner. "A hardware accelerated approach for imaging flow cytometry." In 2013 23rd International Conference on Field programmable Logic and Applications, pp. 1-8. IEEE, 2013.

[10] Remmerbach, Torsten W., Falk Wottawah, Julia Dietrich, Bryan Lincoln, Christian Wittekind, and Jochen Guck. "Oral cancer diagnosis by mechanical phenotyping." Cancer research 69, no. 5 (2009): 1728-1732.

[11] Fedosov, Dmitry A., Huan Lei, Bruce Caswell, Subra Suresh, and George E. Karniadakis. "Multiscale modeling of red blood cell mechanics and blood flow in malaria." PLoS Comput Biol 7, no. 12 (2011): e1002270.

[12] Xue, Feng, Alex B. Lennon, Katey K. McKayed, Veronica A. Campbell, and Patrick J. Prendergast. "Effect of membrane stiffness and cytoskeletal element density on mechanical stimuli within cells: an analysis of the consequences of ageing in cells." Computer methods in biomechanics and biomedical engineering 18, no. 5 (2015): 468-476.

[13] Chahine, Nadeen O., Craig Blanchette, Cynthia B. Thomas, Jeffrey Lu, Dominik Haudenschild, and Gabriela G. Loots. "Effect of age and cytoskeletal elements on the indentation-dependent mechanical properties of chondrocytes." PloS one 8, no. 4 (2013): e61651.

[14] Bao, Gang, and Subra Suresh. "Cell and molecular mechanics of biological materials." Nature materials 2, no. 11 (2003): 715-725.

[15] Di Carlo, Dino. "Inertial microfluidics." Lab on a Chip 9, no. 21 (2009): 3038-3046.

[16] Hur, Soojung Claire, Nicole K. Henderson-MacLennan, Edward RB McCabe, and Dino Di Carlo. "Deformability-based cell classification and enrichment using inertial microfluidics." Lab on a Chip 11, no. 5 (2011): 912-920.

[17] Hou, Han Wei, Ali Asgar S. Bhagat, Alvin Guo Lin Chong, Pan Mao, Kevin Shyong Wei Tan, Jongyoon Han, and Chwee Teck Lim. "Deformability based cell marginationa simple microfluidic design for malaria-infected erythrocyte separation." Lab on a Chip 10, no. 19 (2010): 2605-2613.

[18] Di Carlo, Dino, Daniel Irimia, Ronald G. Tompkins, and Mehmet Toner. "Continuous inertial focusing, ordering, and separation of particles in microchannels." Proceedings of the National Academy of Sciences 104, no. 48 (2007): 18892-18897.

[19] Dudani, Jaideep S., Daniel R. Gossett, T. K. Henry, and Dino Di Carlo. "Pinched-flow hydrodynamic stretching of single-cells." Lab on a Chip 13, no. 18 (2013): 3728-3734.

[20] Stewart, Martin P., Jonne Helenius, Yusuke Toyoda, Subramanian P. Ramanathan, Daniel J. Muller, and Anthony A. Hyman. "Hydrostatic pressure and the actomyosin cortex drive mitotic cell rounding." Nature 469, no. 7329 (2011): 226-230.

[21] Cross, Sarah E., Yu-Sheng Jin, Jianyu Rao, and James K. Gimzewski. "Nanomechanical analysis of cells from cancer patients." Nature nanotechnology 2, no. 12 (2007): 780-783.

[22] Suresh, Subra, J. Spatz, J. P. Mills, Alexandre Micoulet, M. Dao, C. T. Lim, M. Beil, and T. Seufferlein. "Connections between single-cell biomechanics and human disease states: gastrointestinal cancer and malaria." Acta biomaterialia 1, no. 1 (2005): 15-30.

[23] Najarian, Kayvan, and Robert Splinter. Biomedical signal and image processing. CRC press, 2005.

[24] Maini, Raman, and Himanshu Aggarwal. "A comprehensive review of image enhancement techniques." arXiv preprint arXiv:1003.4053 (2010).

[25] Jain, Ramesh, Rangachar Kasturi, and Brian G. Schunck. Machine vision. Vol. 5. New York: McGraw-Hill, 1995.

[26] Parker, Jim R. Algorithms for image processing and computer vision. John Wiley & Sons, 2010.

[27] Liu, Dong Ju, and Jian Yu. "Otsu method and K-means." Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on. Vol. 1. IEEE, 2009.

[28] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms." Automatica 11, no. 285-296 (1975): 23-27.

[29] Al-Bayati, Moumena, and Ali El-Zaart. "Mammogram Images Thresholding for Breast Cancer Detection Using Different Thresholding Methods." (2013).

[30] Kittler, Josef, and John Illingworth. "Minimum error thresholding." Pattern recognition 19, no. 1 (1986): 41-47.

[31] Solem, Jan Erik. Programming Computer Vision with Python: Tools and algorithms for analyzing images. " O'Reilly Media, Inc.", 2012.

[32] Jolliffe, Ian. Principal component analysis. John Wiley & Sons, Ltd, 2002.

[33] Klema, Virginia, and Alan Laub. "The singular value decomposition: Its computation and some applications." IEEE Transactions on automatic control 25, no. 2 (1980): 164-176.

[34] Ledesma, Rubn Daniel, and Pedro Valero-Mora. "Determining the number of factors to retain in EFA: An easy-to-use computer program for carrying out parallel analysis." Practical assessment, research & evaluation 12, no. 2 (2007): 1-11.

[35] Forsyth, David A., and Jean Ponce. "A modern approach." Computer Vision: A Modern Approach (2003): 88-101.

[36] The Scipy documentation for k-means clustering method. URL http://docs.scipy.org/doc/scipy/reference/cluster.vq.html

[37] An article on mean-shift clustering. URL https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/

[38] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24, no. 5 (2002): 603-619.

[39] Ridler, T. W., and S. Calvard. "Picture thresholding using an iterative selection method." IEEE trans syst Man Cybern 8, no. 8 (1978): 630-632.

[40] Trussell, H. Joel. "Comments on" Picture thresholding using an iterative selection method"." IEEE Transactions on systems, man, and cybernetics 9, no. 5 (1979): 311-311.

[41] Rosin, Paul L., and Tim J. Ellis. "Image difference threshold strategies and shadow detection." In BMVC, vol. 95, pp. 347-356. 1995.

[42] Canny, John. "A computational approach to edge detection." IEEE Transactions on pattern analysis and machine intelligence 6 (1986): 679-698.

[43] Gurcan, Metin N., Tony Pan, Hiro Shimada, and Joel Saltz. "Image analysis for neuroblastoma classification: segmentation of cell nuclei." In Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, pp. 4844-4847. IEEE, 2006.

[44] Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. "Contour detection and hierarchical image segmentation." IEEE transactions on pattern analysis and machine intelligence 33, no. 5 (2011): 898-916.

[45] Fawcett, Tom. "An introduction to ROC analysis." Pattern recognition letters 27, no. 8 (2006): 861-874.

[46] Powers, David Martin. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." (2011).

[47] Stehman, Stephen V. "Selecting and interpreting measures of thematic classification accuracy." Remote sensing of Environment 62, no. 1 (1997): 77-89.

[48] The MathWorks documentation for tic function. URL http://www.mathworks.com/help/matlab/ref/tic.html