Automated Segmentation Of Structures Essential To Cell Movement

by

Emma Nicole Fernyhough

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy.

UNIVERSITY OF LEEDS

The University of Leeds

School of Computing

June 2016

Abstract

The study of cells is not only a key field in modern science, but has been an important area of study for hundreds of years. Despite this there is still a lot left unknown. As technology has progressed, so has our ability to photograph and film cells, but much of the processing of these images is still carried out by hand. This is not only difficult and time consuming, but is subject to opinion and error, and often not exactly reproducible.

We are wishing to automate the process of segmenting cells, in order to provide biologists with that data they require to learn more about cells and their movement. This should be done in a quantitative and reproducible way.

Crawling cells, such as those studied for this research, often need to move around the host body, such as the human or other mammal, in order to assist with growth, prevent disease, or to cure damage. To do this they employ other structures which protrude from the cell body to aid their motility. They use very fine hair like features (filopodia) to detect their surrounding, penetrate other cells, and determine direction. They then use thin, flat membranes (lamellipodia) to adhere both at the front and rear of the cell to pull and push forward in the direction of movement.

These features are often extremely difficult to see by eye, making automation of their segmentation an awkward task. To do this, we need to use not only the information in the individual frames of video, but also information gained over time such as their movement between the frames.

We firstly pre-process the images using an automated technique to correct for lighting variations in the footage. Our method is not only extremely efficient and reliable but works equally on different sizes and shapes of cell as well as frames with differing degrees of background coverage, from only one or two small cells in a frame to where the majority of the image is covered. This shading correction method was also tested on non-cellular images taken using the same kind of microscopy to show that it is suitable for all images rather than just those being studied in this work. This pre-processing allows us to make a

simple segmentation of the main cell bodies, which on its own is suitable for cells which do not contain other thin structures.

Using the cell bodies obtained from our pre-processing technique we then find the thinner membranes which are attached to the cell. Despite being a fully automated method, this was more accurate in two out of our three sets of videos than the most popular segmentation program using manual setting of parameters for each video individually.

We improved upon this initial segmentation by incorporating the movement of the cell over time, using an iterative technique to compare the outcome of sequential frames. The result was that our segmentation was better than the manually parametrised segmentation program for every video.

We then wished to find the hair like extensions and again used the information from our pre-processing stage. As these are so difficult to detect by eye we used the information of the movement to create candidate regions where these were believed to be located. Although these were usually not straight, we were able to build up small line segments in the candidate regions to recreate the features and detect the direction.

This allowed us to identify all regions with filopodia present, and to separate them in order to find the required information such as the number, the length, what kind of clusters they grew in and the location compared to the direction of movement. No other method has been found which is able to detect these or segment them separately from the cell.

Acknowledgements

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

Kerrison, N and Bulpitt, A, "Automated Segmentation of Cell Structure in Microscopy Images.", *In Proceedings: VISAPP*, 3 (2014).

Contents

List of Figures

List of Tables

Notation

For the purposes of this work the following notation will be used:

Images - Images will be referred to as a bold, upper-case letter such as $\mathbf{I}$ or $\mathbf{G}$.

Time - Time/frame number will be referred to as a lower-case, italicised t, such that $\mathbf{I}(t)$ refers to an image at time t and $\mathbf{I}(t+1)$ refers to the consecutive frame.

Operations/Functions - Operations on an image will be applied to every pixel in that image such that $f(\mathbf{I})$ implies that the function f is applied over the whole of image $\mathbf{I}$.

Example: $\mathbf{G} = \mathbf{I} > 0.5$ would result in a binary image $\mathbf{G}$ which is the same size as $\mathbf{I}$, and where all corresponding pixels in $\mathbf{I}$ that are greater than 0.5 would have a value of 1 (white) and the remaining pixels set to 0 (black).

Matrices - When a results matrix is referred to by a bold upper-case letter the matrix is the same size as the image matrix it was applied to. **Example:** The edge detection $\mathbf{E}$ of image $\mathbf{I}$ is a binary matrix the same size as $\mathbf{I}$ where the detected edge pixels are stored as a one and the remaining pixels as zeros.

Operators - Binary operators such as $\vee$ and $\wedge$ are applied element-wise across an image.

Chapter 1

Introduction

The study of cells has been ongoing for hundreds of years, since the term was first coined when viewing the holes left by what we now call cells in cork bark. Since then, many progressions have been made, particularly in our ability to store and record footage of cells. Unfortunately, as the amount of data we collect increases so does the time taken to process this data. Much of the information from these images is still collected and processed manually, such as hand drawing the boundary lines and counting features. This is not only extremely time consuming but is also difficult and prone to error. We are aiming towards fully automating the process of image processing and information gathering to provide accurate, reproducible results.

We will discuss the cell footage that this work is based on and the information that biologists wish to acquire from these videos. We will also cover some of the other work which has been done in this area and why this cannot achieve what is required.
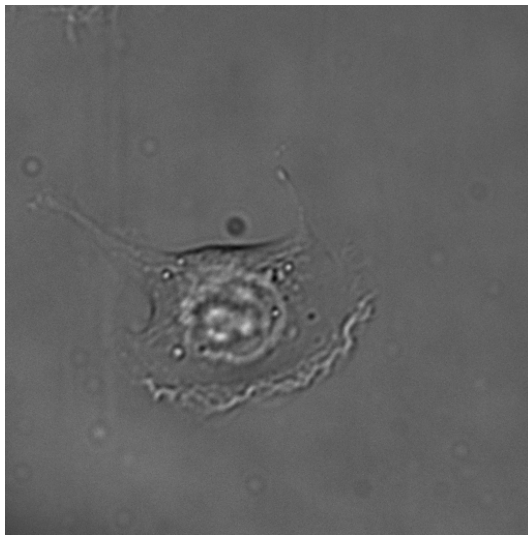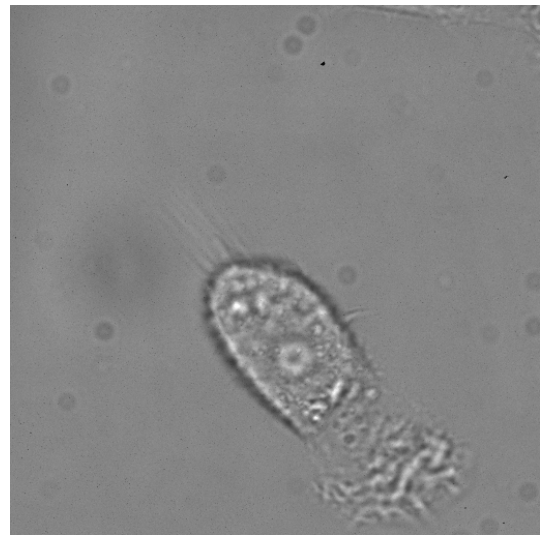
(a) A Cell With Lamellipodia   (b) A Cell With Filopodia

Figure 1.1: **Example Frames From Our Videos.** These two images are frames from some of the videos that this research is based on. The first shows a cell with a spread out membrane (lamellipodia) around the cell body. The second shows a cell with extending hairlike feelers (filopodia).

1.1   Our Cells

There are many types of moving cells, some which swim, others which glide and others which crawl over surfaces. Researchers at the University of Leeds are studying motile cells that employ a crawling motion to move. They wish to know more about how the movement can be enhanced or reduced, and the effect of various chemicals, and expression of genes. Some examples of these cells are shown in Figure 1.1, and it can be seen from this that they are very irregularly shaped. The cells are filmed use time lapse microscopy, with one frame being taken every few minutes, and the shape can vary dramatically over time.

There are two main problems to be investigated. The first is to be able to segment the large flat membranes which the cells use to crawl (Section 2.4), which are called lamellipodia. An example of lamellipodia is shown surrounding the cell body in Figure 1.1(a). The cell body is the brighter, almost circular, object and the lamellipodia is spread

out around most of the body, except for the tail region towards the upper left of the frame. The brighter area towards the bottom right of the cell is where this is 'ruffling' away from the substrate.

Three sets of videos have been taken; one with the cells in their natural state, one set in which they are affected by blebbistatin, and one in which they are affected by brenk. The researchers wish to ascertain whether both the blebbistatin and brenk cells behave differently from the natural cells, and if they behave similarly to each other, in relation to the size and spread of the lamellipodia.

The second problem is to look into filopodia, which are long hairlike structures which extend from the cell, which can be seen protruding as a cluster from the top left of the cell in Figure 1.1(b), which is the leading edge. A singular one is also extending from the right hand side of the cell. These will be discussed in more detail in Section 2.5.

They wish to find if over or under expression of genes can affect the way these grow, finding out if they grow to the same length, in the same types of clusters, and from the same regions of the cells, such as the front and rear.

1.2   Microscopy

Although there are many types of microscopy for capturing cells (Section 2.6), the cells in our footage have been filmed using time lapse differential interference contrast (DIC) microscopy. This is very popular for capturing living and moving cells, as other methods may involve fixing the cells, or staining them which can cause them to move abnormally or to die.

DIC is also very good for filming transparent objects, as it involves splitting a beam of light and capturing the difference after passing through the specimen, creating an image with a 3D appearance based on the thickness of the cells. Some of the issues with this are that it creates a dark and light effect for each cell, so parts of the cell may be both darker

and brighter than the background. It also results in lighting variations in the background itself as the frames appear to be lit from one point, such as a corner or the centre, and will graduate to darker away from this. Examples of this can be seen in Figures 4.1 and 4.2. In the second of these a colour map has been added to aid the visualisation of the lighting change, with the blue being darker and the red being brighter.

As DIC is based on the thickness that the light is passing through it will create stronger features (brighter or darker) where the cell is thickest. This can cause problems when trying to detect the thin features which we are looking for as there will be very little distinction at the edges. The appearance of the inside of large flat objects will also be very similar to the appearance of the background of the frame, so these cannot be found based on the texture.

As the cells we are observing change shape drastically and are very irregular they are not suitable for template matching or for existing models based on the shape.

1.3   The Problem

Although automated methods exist for counting and tracking cells these often only require knowledge of part of the cells location, or feature points within it, and the boundary is not required. Some methods would be suitable were we only interested in finding the cell body, however we are looking to find the parts relevant to the cells motion, of which the body itself plays little part.

Other methods to detect the cell boundary rely on the cell being of a known shape or thick enough to provide a high level of contrast, which is not the case with our cells. The only methods which can detect all parts of irregular cells involve manual activities. In some cases this requires manual segmentation of the area around the cell and of the area it is likely to move to. In others this can need parameters to be set manually. As the parameters change for every video, and sometimes for each frame, the time taken to

optimise these can make the processing extremely slow and laborious. Fine tuning this can also involve image processing knowledge, which may not be possessed by the person wishing to obtain the segmentation.

We wish to create a fully automated system, such that if given a video the output would be a full segmentation of all parts of the cell, with no option setting required. This should work on any DIC videos of cells.

1.4 Our Method

The initial problem tackled was to create a system that can segment all parts of the cell including the thin membranes. The program we created should be able to work equally on all cell types and sizes, regardless of whether they contained lamellipodia or not.

We began by using a fully automated for correcting the lighting variations across the background of the image to allow us to apply other image processing techniques which were not previously possible, such as brightness thresholding. This is applied to all frames which provides us with the basis for all other steps. We can then use simple techniques to segment the body of the cell. From this we built on the information provided to find the thinner areas of the cell if they existed.

Having done this, we then tackled a further problem of attempting to detect the fine hairlike features used as feelers for the cell, again using the information provided by our first technique as the basis to build on for this. No other system could be found which was able to detect these.

1.5 Software Environment

The work in this thesis was undertaken in Matlab [32], which is a high level programming environment. It allows for fast coding of image processing operations by allowing functions to be applied to all pixels in an image.

The image processing toolbox contains many of the popular algorithms used to assist with segmentation and analysis of images. Functions from this toolbox, along with the statistics toolbox, were used during this research.

1.6 Videos

Two main sets of videos were used in this research; the first contained cells in which the lamellipodia were to be segmented and assessed and the second containing filopodia. Both sets were filmed using time lapse DIC microscopy, which is described in more detail in Section 2.6.3. Some more details of these will be given below.

All of the videos were filmed in the DeltaVision format, which is a specialised microscopy format for cell imaging [18], with a .dv extension. They were converted to .avi (Audio Interleaved) format using ImageJ [2], to enable processing by Matlab. The images were captured in 8 bit greyscale, which allows for 256 intensities, with X,Y dimensions of 512×512 pixels.

1.6.1 Lamellipodia Videos

Three sets of lamellipodia videos were captured, with 10 videos in each category. The first were of wildtype cells, filmed in their 'natural' form, with no chemicals added or genes affected. The second and third sets had the addition of blebbistatin and brenk, which biologist wished to compare to assess if both had similar affects to each other and caused the lamellipodia to grow and move differently from wildtype cells.

The videos in the wildtype and blebbistatin categories each had 30 frames and the brenk videos had 40 frames. The time lapse photography took an image roughly every 30 seconds (there was variation between around 28 to 33 seconds).

1.6.2 Filopodia Videos

Two sets of filopodia cells were filmed, with 15 videos in each set. The first set was, again, in the natural wildtype form. The second had genes affecting the cell movement knocked down using siRNA to see if this also affected the growth and spread of filopodia. The images were captured roughly every 120 seconds with 15 frames in each video.

1.7 Thesis Summary

This thesis will be broken into Chapters as are now described. In the next two chapters we will firstly cover some of the background to our problem, including information about cells and their movement, cell shapes, and microscopy techniques. We will then discuss some of the related work in this field, including techniques for correcting lighting variations and methods for automating the process of cell detection for the purpose of counting and tracking the cells, and for finding regularly shaped cells which can be located based on prior knowledge of their appearance. We will also cover research into segmenting the boundaries of irregularly shaped cells, using varying degrees of manual input.

The following three chapters will discuss our methods, broken into the following steps:

Image Preprocessing

We will discuss our method for correcting the lighting of any DIC image, regardless of the size of the objects or the degree of background coverage. We will then show how this assists us in segmenting the cell body from the rest of the image.

Segmenting Lamellipodium

Using the information provided from our preprocessing steps we will describe our iterative technique for segmenting the lamellipodia based firstly on each frame and then corrected and improved using temporal information.

Segmenting Filopodia

We again use our preprocessed frames to segment the filopodia, combining the information from these structures and their movement to firstly create candidate regions and then to find filopodia within them, with the aim of obtaining data, such as length, clustering, and location.

The results obtained from each section will be discussed in that Chapter. The initial system to segment the lamellipodia was created to work as an overall cell segmentation program, suitable for all cell types and sizes. It was tested not only on our videos containing the thin membranes but on other DIC cellular footage.

The addition to segment the filopodia was created as and 'add-on', rather than as part of the original cell segmentation program. This was purely for those cells where we were looking for these structures, although it was also tested on frames where these were not present.

Chapter 2

Background

In this Chapter we will look at some of the background to the work being carried out. We will firstly discuss cells and some of the important factors in studying these. We will then look at cell movement and some of the ways this can occur, focussing on the crawling cells which will be the study of this thesis. We will follow this with how cell shape can affect the speed and movement of crawling cells. We will discuss some of the features of the cell that are vital to movement and why we wish to learn more about those.

We will then discuss ways of capturing cell images using microscopy, including using green fluorescent protein (GFP), phase contrast and differential interference contrast microscopy (DIC). We will cover some of the pros and cons of these and why DIC is preferential for capturing time lapse videos of moving cells.

2.1 Cells

The cell is the building block of life, vital for every plant and animal. It contains the information necessary for survival and heredity. The cell has been studied for hundreds of years since microscopes were invented that first allowed us to see the small compartments where cells had existed. Despite this, there is still a lot unknown about the cell, and as our technology advances so we are able to find more data for analysis [3, 5].

We wish to study not only 'normal', or wildtype cells, but also those which have been affected naturally or by manual techniques. This can assist us in building models or normal movement and growth, which we can use as a comparison to detect abnormal behaviour, helping us to detect which factors are affecting this.

2.2 Cell Movement

Nearly all animal cells show some form of movement, and as such, have developed systems to facilitate this. Even some bacteria have the ability to propel themselves from one host cell into another [50].

One of the simplest forms of cell motility is swimming. This is usually achieved through the rapid movement of cilia or flagella, which they use to 'row'. These cells are greatly affected by the viscosity of the liquid through which they are travelling and use much of the swimming action to counter this [8].

Many other cells do not swim through liquid, but instead crawl over surfaces. This occurs in most animal cells, down to even small predatory amoeba which crawl in search of food. Some cells can crawl to sites of infection as part of the immune system, or to damaged areas in order to initiate repair [3, 8]. Some of the features essential to this will be discussed in Sections 2.4 and 2.5.

The crawling action can range from a smooth continuous gliding to a more irregular, jerky movement with larger changes of direction [3]. This is undertaken by means of

<table>
<tr><td>(a) An Example Of lamellipodia</td><td>(b) An Example Of Filopodia</td></tr>
</table>
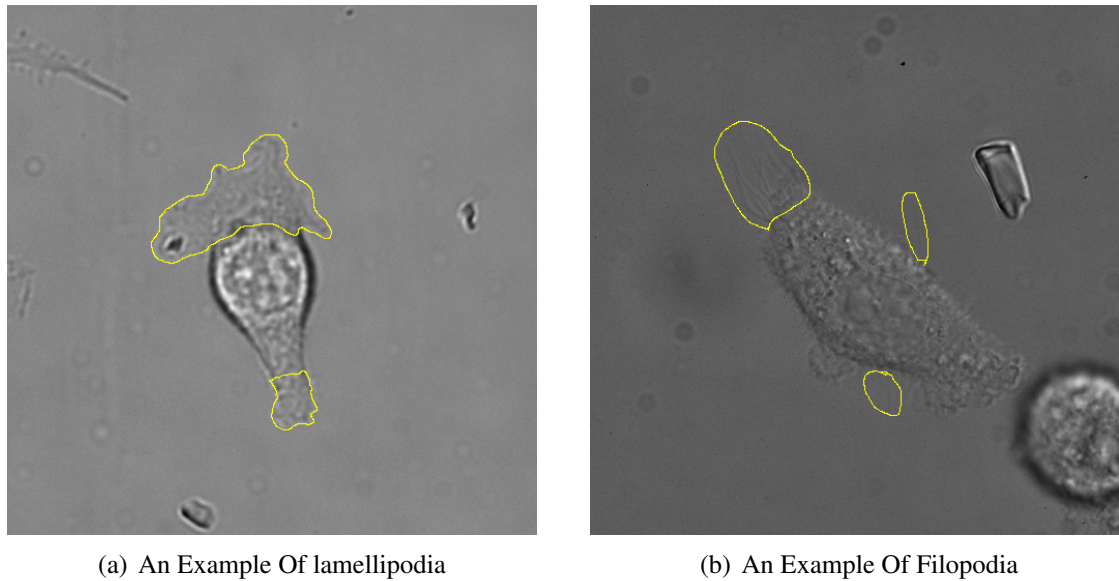
Figure 2.1: **Example Cells Showing Lamellipodia and Filopodia.** The first images shows an example of lamellipodia segmented in yellow. The second shows some regions of filopodia, also surrounded in yellow.

extension of parts or all of the cell, adhesion to the surface, then contraction. This will be covered in more detail in Section 2.3.

2.3 Cell Shape

When a crawling cell, such as those used in this research, is initially introduced to the flat surface on which it is to be studied it will spread from an approximately spherical shape to gradually thin out and flatten to cover an increasing area [34]. Filopodia and lamellipodia will attach to the substrate and allow movement to begin.

In some cells, such as keratocyte cells, movement is fairly simple and the cell changes through four primary shapes. This allows for the shape to be modelled using principal component analysis [39]. They usually move at a constant speed and in only one direction [36]. Although they achieve movement though extension and contraction, the uniformity of their behaviour allows for easier modelling of the locomotive system [19].

In other fibroblasts the cell shape is far more amoeboid and complex. Some examples of the cells used in this research are shown in Figure 2.1. We can see that the shapes vary greatly which causes difficulty in creating a model of these. The cells can also change shape dramatically in short spaces of time. This causes difficulty for segmentation or tracking of the cell parts as templates cannot be used to match with the cells, and features that exist in one frame may not do so in the next [46].

Both the lamellipodia and filopodia are based on actin, and in both this moves backwards towards the cell body as protrusions expand away from it [13]. In Sections 2.4 and 2.5 we will discuss these protrusions, that can be seen in these images in Figure 2.1 and why they are essential to the cell's movement, as well as some of the difficulties in segmenting these.

2.4 Lamellipodia

Figure 2.1(a) shows a cell with lamellipodia at both the top and the bottom of the cell body, the boundary of which has been marked in yellow. Further examples are shown in Figure 5.1, where we can see this occurring in differing sizes and shapes. It can also be seen that without the highlighted boundary the lamellipodia edge is often difficult to detect exactly and that the interior appears to have a very similar texture to the background of the image.

Lamellipodia are large, flat protrusions which should extend from the cell's leading and rear edges. As they expand they stretch out and adhere to the substrate, providing anchor points for the cell to be pulled forward by increasing tension at the leading edge. The cell is also 'pushed' forward by contractions at the rear end [5, 21, 34].

At the leading edge, as the cell begins to move, the lamellipodia lifts from the substrate in a wave like movement, often referred to as 'ruffling'. In Figure 1.1 we can see the ruffling at the leading edge at the bottom right of the frame where it appears as a brighter

white than the rest of the lamellipodia. The ruffles appear to move back towards the cell as the edge of the lamellipodia moves forward. As the cell is drawn forward, the rear of the cell can become a narrower 'tail' until it pulls away from the substrate [8, 34]. Figure 1.1 shows an example of the leading edge towards the bottom right of the cell, and the tail at the rear towards the upper left.

As the cell begins to move forward, the advancing lamellipodia at the leading edge form new adhesions to the substrate, and those at the rear release to allow the tail to move forward towards the body of the cell [13, 40]. The speed of the cell movement can be affected by the strength of these adhesions, as they need to release quickly at the rear end to prevent the cell from being held back [5].

Although it is the lamellipodia at the front and rear that are essential to the movement of the cell, they can also extend and retract unpredictably [8], and grow from other regions of the cell.

Due to the thinly spread nature of these membranes they have very low contrast when filmed using DIC, and therefore cannot be detected based on brightness thresholding. The edges are often extremely indistinct, and do not appear to be continuous to the human eye, so a pixel gradient cannot be found at these points. The irregular, changing shape means that shape models and template matching are not suitable to assist in segmentation and identification.

2.5   Filopodia

Figure 2.1(b) shows a cell with the surrounding areas of filopodia marked in yellow. These filopodia are the fine hairlike structures extending away from the cell. Further examples are shown in Figure 6.1, where it can be seen that these can grow to different lengths and at different angles, as well as in tight groups, more spread out groups and singularly.

Filopodia are very long thin structures, about the same thickness as lamellipodia but

very fine and without the ability to spread sideways as lamellipodia do. They can be produced alone or in addition to lamellipodia. To assist with movement they should extend from the leading edge of the cell, or in the direction of a protrusion, and lamellipodia may grow between them to fill the gaps [5, 34, 35].

The filopodia act as 'feelers', sensing the area around the cell and helping in determine direction [13]. They can grow very long and even penetrate other materials, and seem to have a sensory function [8]. At other areas around the cell, 'blebs' are produced, which appear as highly textured areas of the cell. These appear to reduce as the filopodia extend [34]. These blebs are shown in Figure 1.1(b), where they appear to the bottom right of the cell body.

It can be seen from Figure 2.1(b) that these fine structures are extremely difficult to detect and to count by eye. Although they tend to grow away from the cell they do not grow in straight lines, and are not always parallel to each other or necessarily perpendicular to the cell boundary. They may curve or even cross over other filopodia.

2.6 Microscopy

It has been hundreds of years since microscopes were first invented and then used to study living creatures, and since the term 'cell' was first used to describe the empty holes where what we now know as cells had once been. Despite this, progress in the field has been very gradual and we are still wishing to create new technologies to help us more fully understand the parts of the cell and their roles. It is in the later parts of the 20th century that many advances have been made, and these technologies have become integral to our studies [5, 15, 26].

The electron microscope was developed in the 1930s and provided far better visualisation than was possible with the previously used light microscopes. For the first time we could see cellular detail that was unknown previously. Unfortunately, the fixation re-

(a) Frame From Video Using GFP   (b) Thresholding Based On Brightness
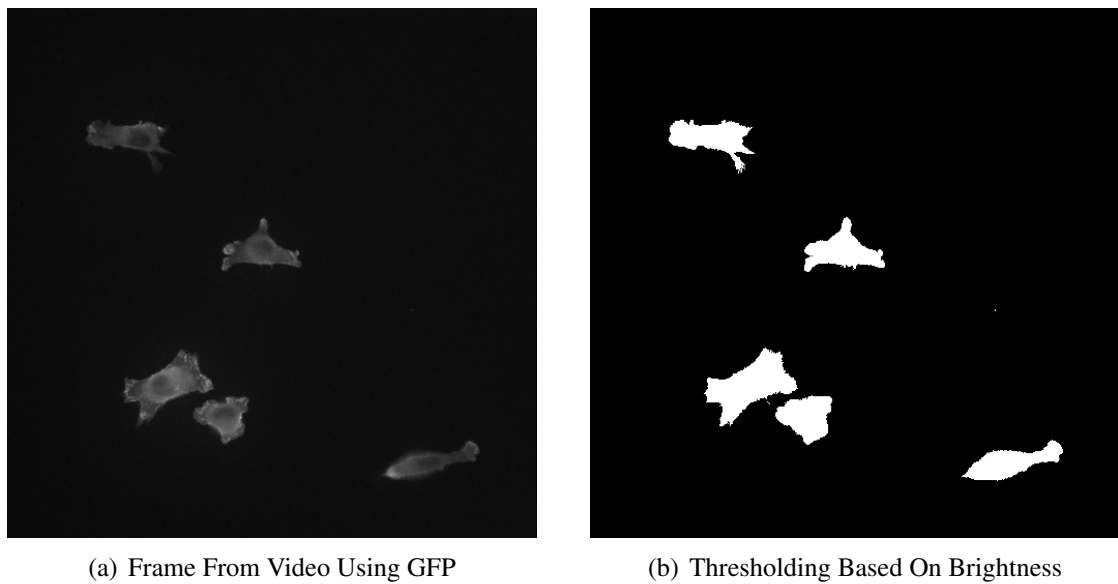
Figure 2.2: **Example of Cells Using GFP.** The first image shows a frame from a video of cells using green fluorescent protein. The second image shows the result of applying a brightness threshold. We can see that the cells have been accurately segmented.

quired to use electron microscopy could not be used on living cells, and it is important when studying motility that we are able to watch the live cell as it moves and changes shape [5, 14, 26] .

Due to this, some of the focus was switched back to light microscopy with advances being made in the 1900s which allow us to access more information than previously. Many of the cells being studied are close to, if not transparent, so methods have been found to visualise these in a way that could not previously be seen by eye. Although staining of cells is possible these can influence motility and cause cell death [15, 26, 48].

We will focus in more detail on some of the most popularly used microscopy techniques for live cell imaging in the following sections, including cells marked with fluorescence, phase contrast and differential interference contrast.

2.6.1 Green Fluorescent Protein

Green fluorescent protein (GFP) was originally discovered and purified from the *Aequorea victoria* jellyfish in the 1960s, and was then cloned to be used as a marker for microscopy. It was found that this could be used to highlight specific molecules and proteins within cells that would fluoresce under a blue light [3, 15, 26].

This ability to make the cell 'glow' provided us with vital information and aided clarity of visualisation. Due to the bright nature of the glow against an unlit background, segmentation of a cell or its structures could often be undertaken purely with thresholding based on the brightness of the pixels. An example of this can be seen in Figure 2.2 where we have shown a frame from a video filmed using GFP. We then show the effect of applying a brightness threshold to the frame. All parts of the cell have been segmented accurately from the background, with the binary result showing the cells in white and the background in black.

GFP progressed away from using just the 'green' protein, and is now available in many colours so that individual parts of the cell can be labelled individually. This allows for more indepth knowledge of, for example, when particular genes are expressing, and can allow us to track individual substructures in the cell [52].

Unfortunately, adding the GFP to the cell affects its behaviour. The cells may move differently, and in fact light can be used to force parts of the cell to move in a way it would not normally do [3]. The GFP can cause damage to the cell, as can prolonged exposure to light [52]. This can cause premature cell death, which was observed in the video used for the example in Figure 2.2, one of the cells rounds up and dies half way through the filming, and by the end none of the cells are moving.

As we wish to study the cells moving over time we need to find a method that does not harm the cell, that involves minimum illumination, and that allows the cell to move as naturally as possible.

Transparent Specimens in Phase Contrast and DIC
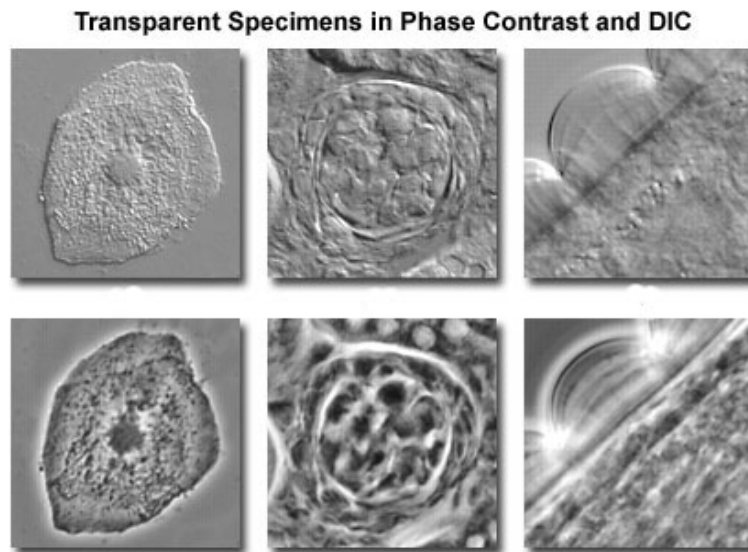
Figure 2.3: **Comparing Phase Contrast to DIC.** These images on the top row show specimens imaged using DIC microscopy and those on the bottom use phase contrast [47]. The phase contrast images show the glowing 'halo' effect.

2.6.2 Phase Contrast Microscopy

Phase contrast microscopy was invented in the 1930s and uses individual rays of light passed through the specimen. As the light is disturbed by the thickness of the object the velocity changes and the resulting pattern gives a contrasting image of dark and bright regions. This can be useful as it uses the different thickness of a cell to visualise it, even when it is transparent [5].

Although phase contrast is ideal for studying living cells, the resulting images show a bright 'halo' around the thicker areas, such as the cell body. This makes it difficult to detect the cell shape with accuracy as the edges are indistinct. It can also be unsuitable for thicker specimens, such as the rounded cells that occur when a cell is dying, as these will glow with a brightness that affects the rest of the image, and therefore affects brightness thresholding techniques. Due to the nature of this type of microscopy, some of the same structures in the cell will appear as bright objects while others appear to be dark, making it difficult to identify similar objects [3, 5, 26, 48].
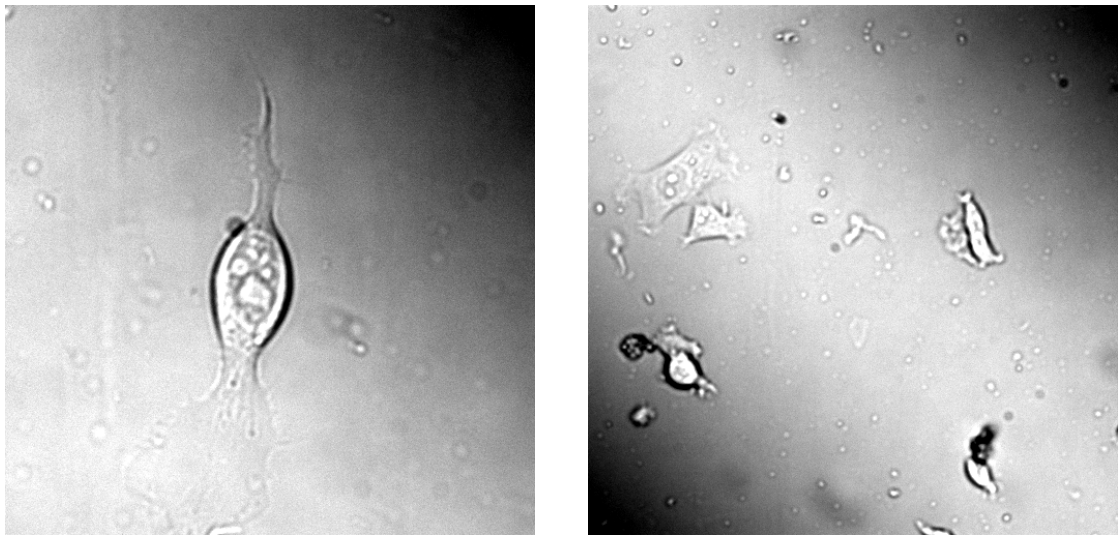
Figure 2.4: **Example Images Using DIC Microscopy** These images taken using DIC microscopy show the characteristic shadow effect giving the cells a 3D appearance, as well as the lighting variations across the background.

The images in Figure 2.3 (taken from [47]) show a comparison of phase contrast and DIC images. The bottom row shows the phase contrast images with the characteristic halo effect. It can be seen that this makes the edges less distinct than in the DIC images. The DIC images on the top row, which will be discussed in Section 2.6.3, show clearer edges. The first image in particular shows the cell standing out from the background as if in 3D.

2.6.3   Differential Interference Contrast Microscopy

Differential interference microscopy (DIC) was invented by Nomarski in the 1960s to provide higher contrast images than was previously possible, and in higher resolution than other light microscopy. It is similar to phase contrast microscopy, but uses a prism to split the light beam into two rays. As with phase contrast, the ray is more affected by the thicker parts of the cell, but with DIC the image is recreated based on the differences between the two rays as they recombine [5, 26, 38, 45].

The resulting images from DIC microscopy show a psuedo-3D effect caused by the dark and light sides of the cell. Some examples of this can be seen in Figure 2.4, where

we can see the 'shadow' effect. One of the problems with DIC is also shown in these images, where not only are the cells showing shadow effects but this is also shown across the whole frame. This will be discussed further in Chapter 4, as removing this can assist with other image processing techniques. Although it is recommended to take an empty background frame to use to rectify lighting inconsistencies this is was not available with the footage used for this research. Additionally, we wish this method to work on any DIC cell footage so this should not be a requirement [5, 16, 26].

Another issue that occurs is that, as the cell is shown as if lit from one side, it is often dark on one side and bright on the other. The result of this is that the cell border and internal substructures will be both darker and brighter than the surrounding background [48]. Due to this we cannot simply apply brightness thresholding, as was done in Figure 2.2.

As DIC is not an invasive technique, it is ideal for filming living cells over a period of time [28]. Unlike phase contrast, it works successfully on thicker samples, but will produce lower contrast results on thinner areas. A comparison of the results from both phase contrast and differential interference contrast is shown in Figure 2.3. Large spread out areas, such as lamellipodia (Section 2.4) will show a very similar appearance to the background of the frame, and very thin areas such as the filopodia (2.5) will not cause a large change in the light so will not show a high contrast.

2.7   Summary

As we are wishing to film moving cells over a period of time we need a microscopy method that will not damage the cell or affect its movement. Electron microscopy is not suitable as the cell would need to be fixed, and staining or using GFP can cause abnormal movement or cell decay and death.

Both phase contrast and DIC are suitable for filming living, motile, transparent cells.

As we are wishing to find more detail about the cell and its shape, DIC is preferential to phase contrast as it is higher resolution and does not have a halo effect causing indistinct edges. To film live cells we require minimal damage, but this causes an increase in noise, and therefore causes more problems with image processing techniques.

Although phase contrast and DIC produce images which make the transparent more visible, neither are ideal for using computers for image processing due to inherent lighting effects and low contrast. Additionally, DIC causes lighting inhomogeneity across the entire frame which affects many other operations.

Chapter 3

Related Research

This chapter will discuss some of the previous research into the processing of cellular images. It will firstly discuss the approaches which have been used to tackle the problems inherent in DIC imaging, such as the lighting variations and the lack of contrast. It will then investigate methods used to automate the process of detecting and segmenting cells, as well as analysing the shape of these, particularly during movement.

We will discuss why many detection techniques are not suitable for our purposes as they have been created to obtain other information, such as tracking the cells, and do not provide the boundaries and segmentation that we require.

3.1   Lighting Intensity Variations

One of the issues with images taken using DIC light microscopy is the intensity inhomogeneity, in that the image appears as if it is being lit from one location, which can vary on a frame by frame basis. Examples of this can be seen in Section 4.1, which shows some

examples of images which appear to be lit from one corner and are therefore darker in the other corner (Figure 4.1 (a) and (c)), an image which appears as if lit from the top of the frame (Figure 4.1 (d)) and also an image which seems to be lit diagonally across the middle, and is therefore darker in two of the corners (Figure 4.1 (b)).

This problem is discussed by Tomazevic *et al.* [54], who describe how this often has very little impact on manual analysis of the images, but would interfere greatly with automatic analysis. They describe how a good method for resolving the problem would resolve any shading issues, if present, but would also not affect a non-shaded image. If there is no ground-truth (i.e. an unshaded image) for the real images, as in our case, they state that the best approach to evaluate a method is to use artificial images, some of which are shaded and others of which are shading free. This is the technique we use in Section 4.3.2 to evaluate our method.

Figure 3.1: **Cell With Thin Lamellipodia and Thicker Body.** This cell shows a thin, spread out lamellipodia extending from the top edge, but a thicker body with more obvious sub-structures.

Li and Kanade [28] discuss how they precondition DIC images of stem cells and

blood cells in order to increase the segmentation accuracy. They propose two methods, each of which is suited to different types of images, one for those including small smooth objects and an alternative for large, flat and blocky objects. This is not particularly suitable for our images which can contain objects of different sizes, as well as different textures within each object. Although the lamellipodia are large and flat they are also thin and smooth rather than blocky. These are attached to the cell bodies which contain many small parts which are not smooth. An example of this can be seen in Figure 3.1, where we can see a thin lamellipodia spreading from the top of the cell which is attached to a thicker body with more obvious structures that are showing as brighter and darker than the lamellipodia, which shows very little contrast difference from the background. We are hoping for a single method which can convert the intensity over any DIC image, regardless of what it contains and with no loss of any information.

ImageJ [2], which is an image processing program for scientific images, includes plug-ins designed to perform background illumination correction. All of the methods make assumptions about the image characteristics and often introduce image artefacts. The method most suitable for achieving an even background illumination without an empty background slide, taken before introduction of cells to the microscope, to subtract from the resulting videos is the 'pseudo flatfield' method. This method also filters the image using a kernel, but uses the mean of the region selected. The implementation of this involves manual parameter setting for each image, based on the size and nature of the contents, rather than being fully automated.

We were unable to receive very accurate results using ImageJ as the larger size kernel that was used, the larger the size of a dark shadow was left in the most shaded part of the image. Changing the parameters to use a smaller kernel resulted in parts of the cell also being blurred so there was a loss of clarity, particularly at the boundaries of the cells. Results of using this on a manually created image are shown in Section 4.4.

3.2 Segmentation and Shape

This section will discuss some of the work done on automation of cell detection, such as for the purposes of tracking or counting cells. It also discusses analysis of cell shape, particularly for motile cells, and the methods used to capture this information.

3.2.1 Detecting Cells

When studying normal and abnormal cell movement it is important to not only find the location of the cell but to segment it in such a way that the boundary can be clearly defined to provide us with information about the about the size and shape of the cell. Much of the previous work on cell detection has been for the purpose of tracking or counting cells, which requires only part of the cell or a single feature to be detected. This approach was taken by Li *et al.* [29] who created an automated system capable of tracking hundreds of cells simultaneously, captured under phase contrast microscopy.

Ray and Acton [41] and Sirakov *et al.* [49] both used active contours to automate the tracking of cells by obtaining a rough estimation of the shape. The accuracy of the shape was not measured as this was not relevant to the research as the goal was to evaluate the accuracy of the tracking. Level sets were used by Mukherjee *et al.* [37] to detect and track cells in an automated manner. Their segmentation used a template to find cells which were roughly circular or teardrop shaped, but would not be suitable for cells with an irregular shape. Li *et al.* [29] again fully automated cell tracking by finding large parts of the cell but not the edges in phase contrast microscopy.

Jiang *et al.* [20] also worked on DIC microscopic images and found that due to the low contrast achieved it was preferable to use SIFT points to follow the cells between frames. The cells they investigated changed shape dramatically between frames which was not suitable for model based tracking. They found that as the border of the cells was very vague it would be easier to track the points rather than segment the cells from the

background.

CellTrack [44] is a popular piece of open source software for tracking cells. Initial cell boundaries are found using Canny edge detection which involves amending the higher and lower thresholds to link the edges and changing the number of dilations and erosions until the boundaries are more accurately selected. The cells are then filtered by setting a maximum and minimum size to remove non-cellular material in the images. The same settings can be used across a video or the parameters can be set for each frame individually to improve accuracy.

Cells can then be tracked by a choice of methods, including template based tracking, which looks for rectangular patches surrounding the cells then finding overlapping rectangles in the following frame, centroid tracking, and optical-flow tracking, which can cause irregularity with the boundaries and 'leaking' of intra-cellular features [42].

With time spent on amending all of the necessary parameters, this software can produce a segmentation of the cell boundaries and provide details of the overall cell size. Due to this, CellTrack was used as a comparison for our automated results in Section 5.5 and Section 5.7.

3.2.2   Cell Shape

Pincus and Theriot [39] compared methods for cell shape analysis, evaluating techniques that could record the shape without introducing additional information and that could capture biologically relevant variations in the shape. They found that principal component analysis (PCA) was the method that can be used most accurately to capture modes of shape variation. PCA has been used very successfully on keratocyte cells, which can be modelled with only a few modes of shape variability. This method, which works well when the cells move through a small number of pre-known shapes, is not suitable for more amorphous structures where the features cannot be aligned [22].

Much of the previous work on cell shape segmentation has involved some degree of

manual segmentation, which is very time consuming. Wu *et al.* [57] wished to segment unstained cells filmed with light microscopy using a fast, robust method. Their approach involved an operator not only manually locating the cell but also determining a surrounding window in which the cell would be likely to move, based on previous experiments. To ensure that only one cell was segmented they dismissed those cells which potentially had overlapping tracking windows. Using this technique allowed them to lessen the effect of intensity variation which appears across the whole image and allows for local thresholding. This research only discusses cells which appear small compared to the size of the frame and not those which cover larger parts of it and therefore would cover differing intensities.

Bise *et al.* [6] looked into not only tracking but also studying shape deformation in cells in DIC images. They preconditiond the images to find the central area of the cells, excluding protruberences, and could then identify each cell individually even when it was close to another. Although this provided some shape information necessary for identification, many parts of the cell were excluded from the segmentation such as the long thin parts and those which deform significantly.

A two-step approach was used by Kuijper and Heise [25] to automatically segment cells in DIC images. The first step was to detect the centres of the cells using a voting method over the detected structures, then to use these as seed points for detecting the boundary using PDE's in a level set framework. It involved the setting of one parameter to determine the boundary condition based on the cell centres. They received good results as long as the cells were not too clustered and did not contain obvious sub-structures. This method also assumed that the cells being observed showed symmetry and only a rough estimation of size was necessary.

Young and Gray [58] used edge detection on their DIC images to segment cells of differing sizes and orientations as they could find no fully automatic method. They also relied on the cells being circular or elliptical, with a smooth contour. Although they were

able to identify cells, even when touching or overlapping, the outlines on real images were often irregular

Investigation into segmentation of irregularly shaped cells was undertaken by Simon *et al.* [48]. After altering the image to account for the lighting variations they then thresholded the image and outlined the cell using the largest thresholded area. Although this was capable of segmenting cells with a complex shape they had problems with frames that had a high ratio of background to cell and cells with poor intensity contrast. They found that this often occurred with cells with thin membranes and so excluded these from the segmentation process.

Brannock *et al.* [7] used wavelets to detect filopodia as this was found to be more succesful than other edge detection methods. They tested against the filopodia of biological significance, which appeared to stand out clearly in the images shown, unlike in our videos. They were able to show some of the regions containing the filopodia, but were not able to segment them individually. This method scored positively for regions where filopodia where detected against manual images, but did not count areas that were incorrectly identified.

Wenzel [56] also wished to detect filopodia, but as these occurred around full circumference of the cells being studied it involved creating a larger boundary for the cell rather rather than separating the filopodia. In both of these methods the area not containing filopodia, such as the gaps in between, is not used to measure the accuracy of detection, therefore we have no knowledge of the false positive rate.

3.3   Summary

We have discussed some of the research aimed at automating the process of detecting cells in microscopic images, particularly light microscopy. Many of the automated methods proposed are for the purposes of either tracking cells or counting cells, and as such the

cell boundary does not need to be detected. Some methods for detecting the cell boundary involve levels of semi-automation, such as selecting the area around the cells manually.

The methods which can undertake full automation of the detection of the cell boundary often rely on knowledge of the cells shape or size and will not work on complex cell shapes. The only automated detection found for such shapes cannot detect thin membranes or cells with low contrast to the background.

We aim to provide a fully automated method with no manual setting of parameters that detect all parts of a cell in DIC images, including thin membranes and long protruberences which are vital to a cells movement. This should be equally successful on images of multiple cells, those with large areas of background, and on images of single large cells which can cover a high proportion of the images. No prior knowledge should be needed regarding the shape or nature of the cell and there should be no necessity for a background frame to account for issues with the lighting.

Chapter 4

Image Preprocessing

In this chapter we will discuss some of the processing which we apply to the images prior to segmenting the difficult to detect parts of the cell used for movement. In order to perform other techniques more effectively we need to remove the problem of uneven lighting caused by DIC microscopy. We will describe our efficient solution and show results not only on the cell images being studied for this research but also on unrelated DIC images. We then apply thresholding to the resulting images and from the results show the improvement on using this technique on these compared to unrectified images. We will compare these results to those obtained by the best currently available methods for removing the lighting effect.

Further to this, we will additionally apply edge detection to the images and combine this with the thresholding to show that this detects parts of the cell with greater accuracy than other methods.

4.1 Lighting Problem

As discussed in Chapter 2.6.3, DIC microscopy causes a shadow effect on the images, as can be seen in Figure 4.1. The first image (Figure 4.1(a)) shows that the frame is darker in the top right, gradually getting lighter towards the bottom left. The second (Figure 4.1(b)) shows the top right and bottom left of the frame to be darker with the brightest area in the middle. It can be seen that the lighting variations do not just affect the background of the image but also affect the cell, as can be seen in Figure 4.1(a), where the top of the cell is in a shaded area and appears much darker than the bottom of the cell, which is in a brighter area.

The additional two images are from an online image search. The first shows a group of cells which cover a large part of the frame with very little background showing (Figure 4.1(c)) [51] and the next (Figure 4.1(d)) is a DIC image of a snowflake [17], included to showing the lighting variation on a non-cellular image.

Due the variation in lighting intensities, brightness thresholding on the images is not possible as they are. Results of this are shown in section 4.5. The background varies from very dark to very light, so parts of the cell will be brighter and parts of the cell will be darker than the background. It is not only the background which is affected but the cell itself also has intensity variations depending on where in the frame it is located. These variations can be seen more clearly in Figure 4.2 column(a), where a colour map has been added to the pictures to enable the variations to be more apparent.

We need to be able to correct for this lighting disparity in such a way that there is no loss of detail from the original image, but also ensuring that an unshaded picture would not be affected.

(a) Single cell with brightest light in the bottom left
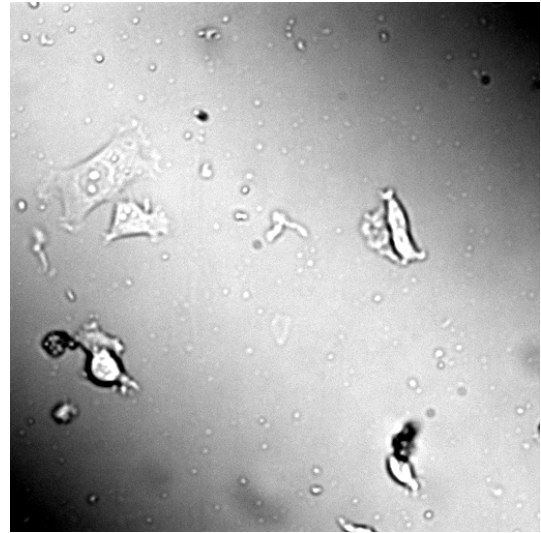
(b) Multiple cells with the shadows showing in the corners
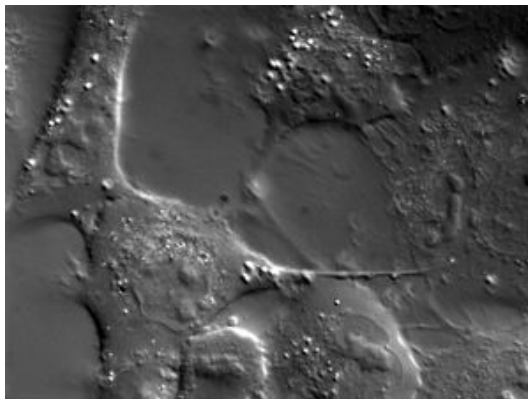
(c) Large cells covering much of the frame with shadow in top right [51]
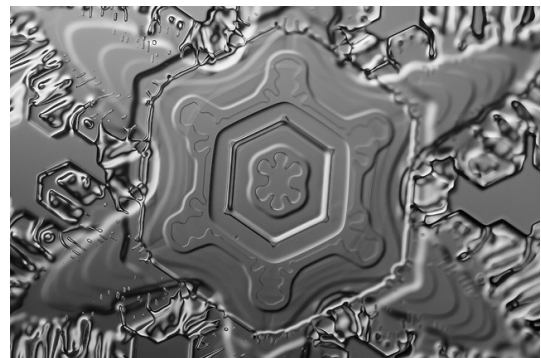
(d) A snowflake which is darker towards the bottom of the image [17]

Figure 4.1: **Images Showing DIC Shadow Effect.** These images show the lighting variations in DIC images which appear as if the image is lit from one area and is in shadow in the others.

4.2   Correcting the Lighting

The first step of our process to resolve the shade variations is to apply a large Gaussian blur to the image, $\mathbf{I}$, to give a filtered image $\mathbf{G}$. We have shown $\mathbf{I}$ in Figure 4.2(a) in it's actual form. In Figure 4.2(b) we have artificially mapped this to the 'jet' colour map in Matlab, which allocates colours from blue (0/black), through green, yellow and orange, to red (1/white). This has been done to make the shading easier to visualise. To automate the process the filter is set to be a square about a third of the width of the image size. The size was chosen to be large enough to remove all detail in any image without need for manual parameter setting.

We can see the effect of this in Figure 4.2 column (c), which shows the lighting variations across the frame but is no longer showing any parts of the cell or object being studied. This is effectively providing us with the kind of information that would be obtained if we had access to an a priori frame, taken to show the camera and lighting effects without including any cells. We can clearly see how the shade varies smoothly across the image, without any need to model the background.

We find the difference ($\mathbf{D}$) between the original image and the filtered image with the Gaussian blur applied, so that the difference $\mathbf{D} = \mathbf{I} - \mathbf{G}$. Finding the pixel value difference between the blurred and the original image provides us with all the information necessary relating to the 'detail' of the cells and other objects in the image. The background of our frame is then set to the mean of the Gaussian blurred intensities, (a scalar value which is the mean of all pixels over $\mathbf{G}$), which we will denote as $\mu_{\mathbf{G}}$). This is done as some of the values in our difference image are negative and some are positive, therefore we require a value to centre these around so that they fall back in the range of zero to one. The 'detail' is added to this background, so that the new image $\mathbf{I}' = \mu_{\mathbf{G}} + \mathbf{D}$. This very fast process is repeated for each frame in the video.

The corrected images, $\mathbf{I}'$, are shown in Figure 4.2 column (e), and again in column (d), which has once again been mapped to the 'jet' colour map, as describe above. It can

(a) Original (**I**) (b) **I** (Artificially Mapped to a Colour Space) (c) Filtered Image (**G**) (d) **I**′ (Artificially Mapped to a Colour Space) (e) Corrected (final image **I**′)
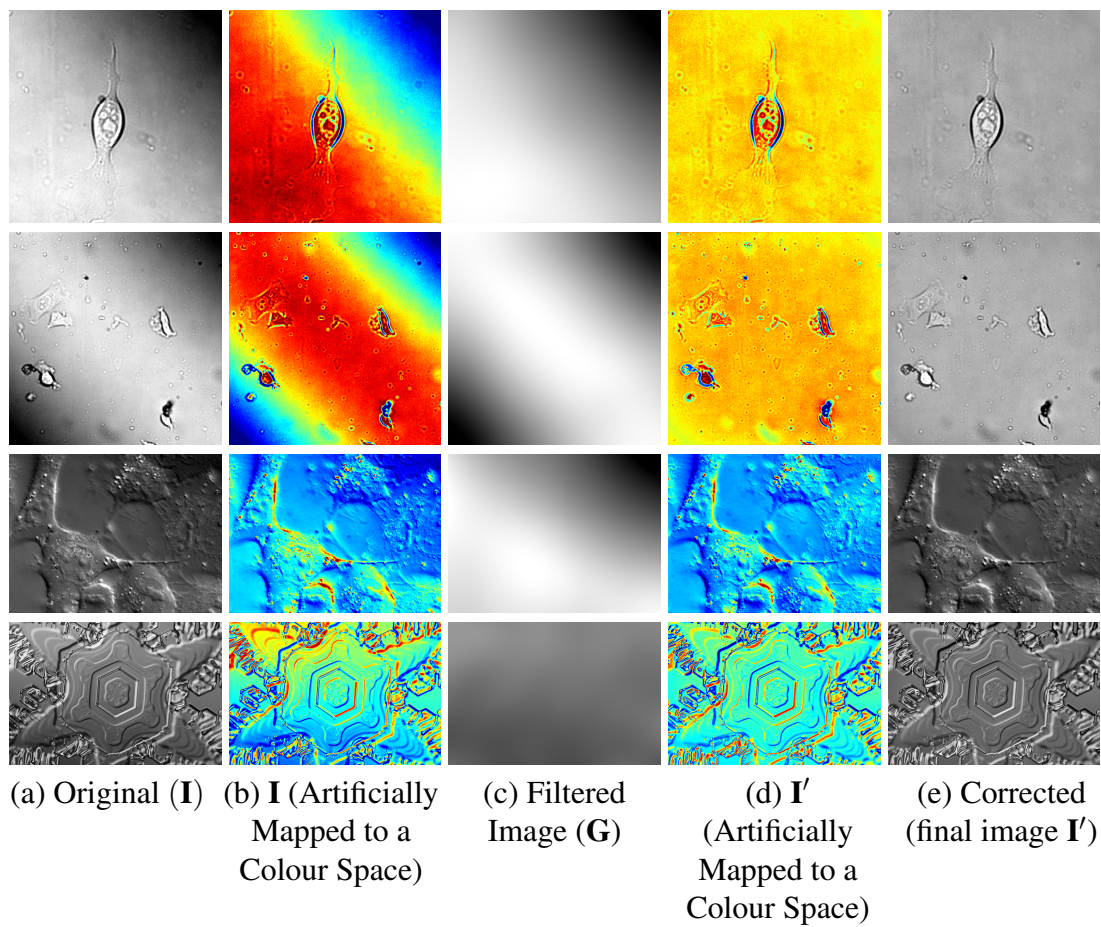
Figure 4.2: **Images Showing Lighting Correction.** These original images are shown in columns (a) and (b). Column (c) shows the shadow effect without any cell detail. Columns (d) and (e) show the images with a level lighting across the background.

be seen in every case that the full detail from the original frame is showing but that the background no longer has a lighting variation.

As this pre-processing step is run on every frame before any other steps are carried out, for the remaining chapters of this thesis when we refer to the image, or frame, **I** we are referring to the resulting image from this process **I**′.

4.3 Results

We will firstly look at the results of the method on the images shown above (Figure 4.2), two of which are from our video and two from an online search, to show that the method is equally effective on any image captured using DIC microscopy. We will then apply the method to an image which has been manually shaded so that we are able to compare the results with the original.

Although the results in Section 4.2 show the visual changes achieved, we also need to show the values obtained. We will use different methods for this, as the images above do not have ground truth images, so we will use the pixel values across the line of greatest change to show the 'intensity profile' [54] before and after our method. We will then use direct comparison of resulting images and the ground truth on the manually created images.

4.3.1 Testing On Real DIC Images.

Although the shading correction can clearly be seen (Figure 4.2) by visual inspection, we have no ground-truth (e.g. no unshaded image) to compare the corrected images with. To show the effect of our method on the values across the image we have taken the measurement across the line with the highest variation and shown these in Figure 4.3(a). This shows that although there are peaks and troughs for the dark and light aspects of the cell, caused by the DIC microscopy, there is also an underlying trend for the pixel values to increase or decrease as the line progresses from the lighter to the darker corner or side.

We then calculate these again after the Gaussian blur (Figure 4.3(b)). This shows only the shading of the image without any detail. It is easier to see from these charts which of the images have the most drastic lighting changes and also whether this is only in one direction or in more than one. For example row 2, which shows that image progresses

(a) Original Image **I**	(b) After Gaussian Blur **G**	(c) Corrected Image **I′**
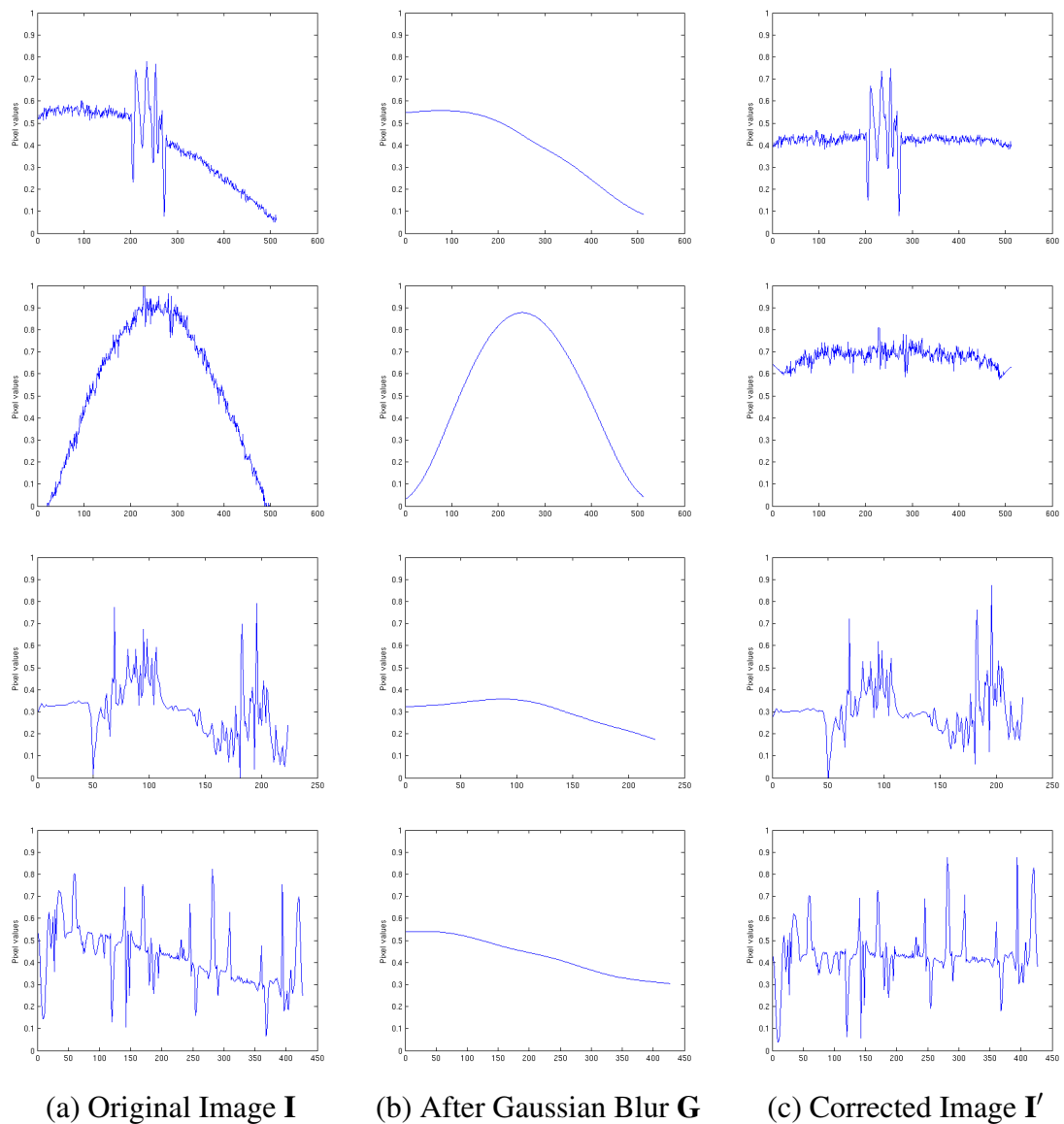
Figure 4.3: **Pixel Values Before and After Correction.**
The pixel values were recorded along the line with the most lighting variation, which was the bottom left to top right of the first three images and vertically for the fourth. These are shown before and after lighting correction, and also with just the shading visible after the Gaussian blur. The images are those shown in Figure 4.2. The pixel values all show a more even background without loss of information of the object being captured.

from dark in one corner, to light in the middle, then dark again in the opposite corner, whereas the final row, calculate from top to bottom, shows only a gradual decrease in brightness
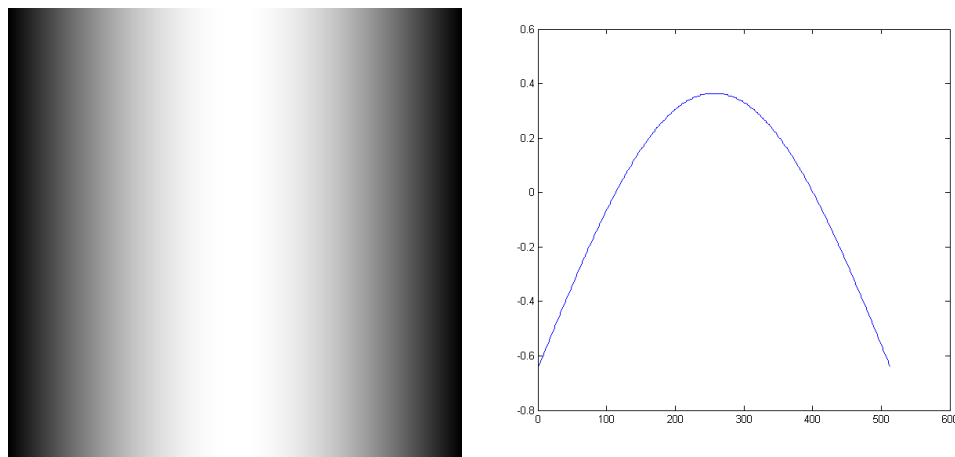
Figure 4.4: **Manually Created Shading/Lighting.** The first image shows the created sinusoidal shading and lighting using. The second shows the profile of this after removing the mean.

We finally show the pixel values along the same line after applying our method (Figure 4.3(c)), which shows the peaks and troughs of the detail has not been lost but that the background level is now more even.

4.3.2 Testing On Artificially Shaded Images

We would like to be able to show that this method meets the necessary criteria that it will have no effect on an unshaded image as well as correcting the problem accurately on a shaded image [54]. To do this we have created a manual 'ground truth' image [31] and ensured that this is free of shading by setting all background pixels to be the same value.

We then create a manual shading and lighting, as shown in Figure 4.4(a), by repetition of a sin wave the length of the image. To do this we created an image the same size as **I**, which we will denote as B, then for each pixel in the ith column,

$$B_i = sin(x_i), \quad \text{where} \, 0 \leq x_i \leq 180 \tag{4.1}$$

As this ranges from zero to one, we remove the mean from this to ensure that some

of the pixels are increased and some are decreased. The profile of this is shown in figure 4.4(b). The resulting matrix is added to the manual image.

The original image with the manually adjusted background is shown in Figure 4.5(a) and the shaded and lightened image is shown in Figure 4.5(c). We ran our method to correct the lighting variation issues on both of the images and the result for the image with an even background is shown in Figure 4.5(b), and the result for the image with the manual lightening and darkening is shown in Figure 4.5(d).

To check the accuracy of our results we compared the pixel values on the resulting image to the original unshaded image. With pixel values between 0 and 1, the largest difference over both of the resulting images was less than 0.0027. With 256 grey levels, this would be less than 0.7 of a shade, so when rounded would show at most as one pixel level out. The most likely reason for any differences are that the original images have the values stored up to two decimal places, but the calculations in the method are done to four decimal places.

This shows that we can accurately remove any variations in shading intensity across an image with absolutely no loss of the detail contained, and can also run our technique on an image with no lighting variations without changing the original image. This is all done with no manual parameter selection as the technique calculates the relevant kernel size based on the size of the image.

One of the main motivations for amending the lighting is to ease the process of applying other image processing techniques, such as thresholding, which is not possible on the unrectified image, which we will discuss and show in Section 4.5.

4.4 Comparison With Alternative Method

ImageJ [2], is an open source image processing program for scientific images, as discussed in Section 3.1. This offers a plug in for resolving lighting variations in microscopy
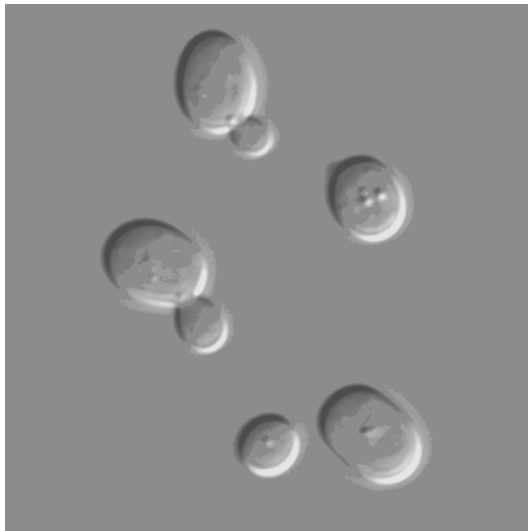
(a) Original, unshaded image [31]
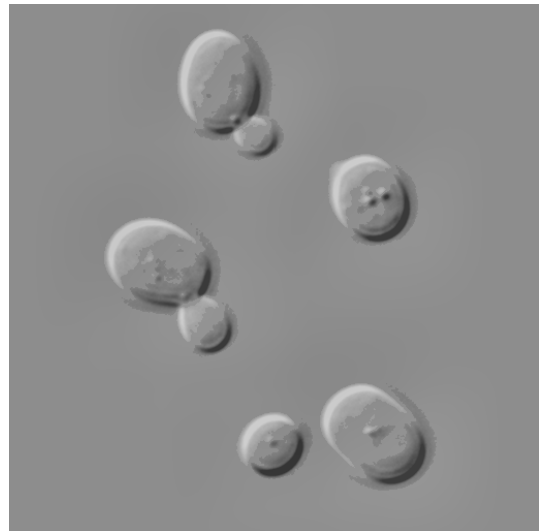
(b) Result of running our method on image (a)
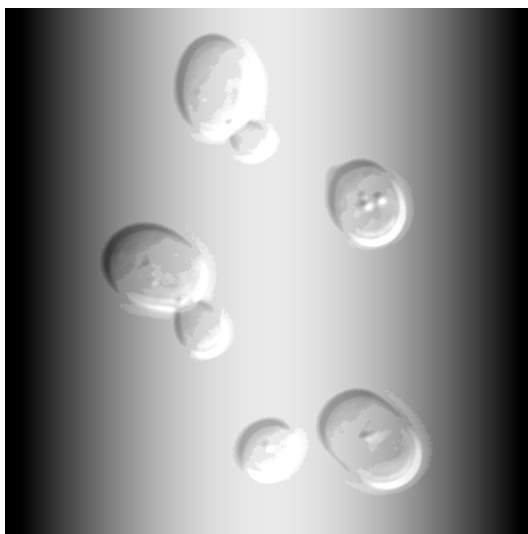
(c) Image (a) with shading and lighting manually added
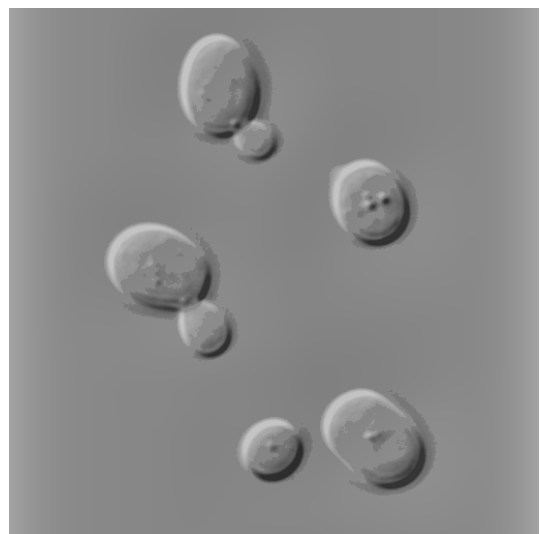
(d) Result of running our method on image (c)

Figure 4.5: **Images Showing Results on Test Images.** The images show the results of testing our method on an unshaded image and on an image with shading manually added. After our method the resulting image from both tests was exactly the same as the original image, proved by subtracting one image from the other.

(a) Original Image   (b) Kernel Size 10   (c) Kernel Size 150

(d) Manually Shaded Image   (e) Kernel Size 10   (f) Kernel Size 150
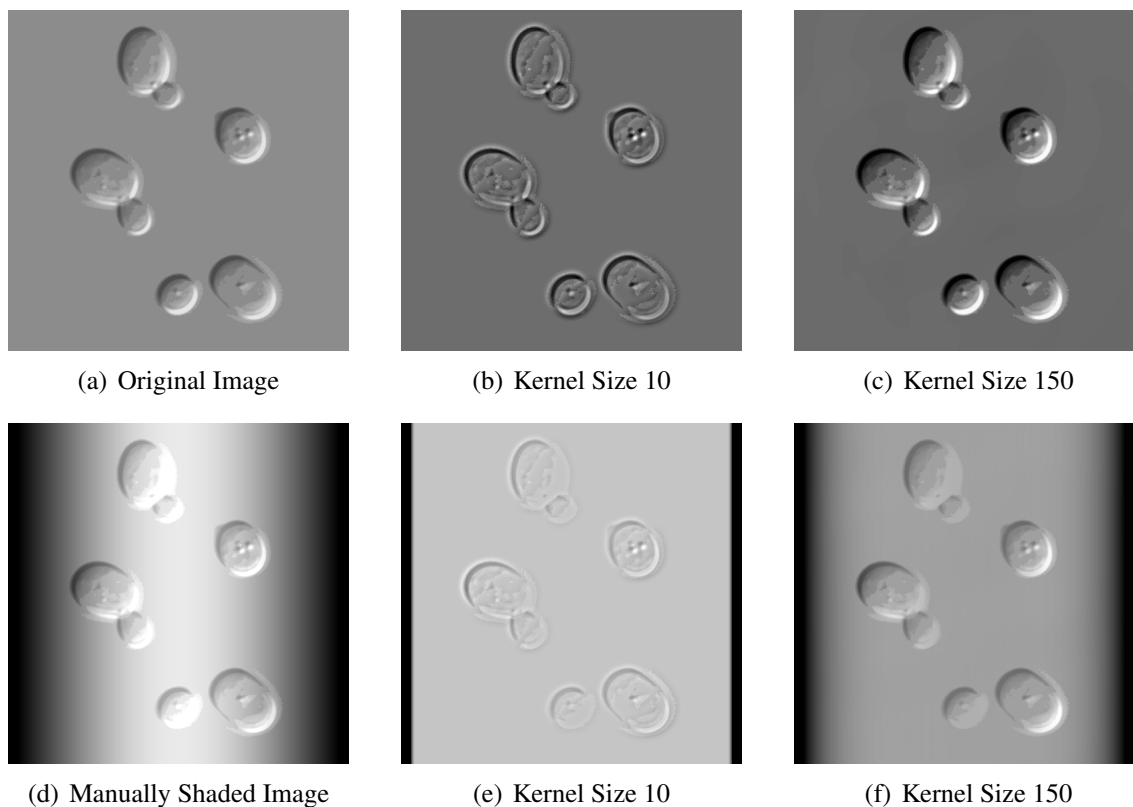
Figure 4.6: **Results of Alternative Method on Manually Created Images.** The first row shows the image with no shading and the results of running pseudo flatfield [10]. With both kernel sizes the background is darker, and with the smaller the cell has also lost definition. The second row shows the manually shaded and lightened image. Both kernel sizes leave a dark shadow at the sides, but this is more pronounced with the larger.

images where an empty background slide is not available. The Pseudo Flatfield [10] method involves manually setting the kernel size parameter. This is then used to set the pixel values to the mean of a window of the size of the kernel to create the flatfield. This is then removed from the original image and the resulting pixel values adjusted to be in the 0 - 255 range.

The results of running pseudo flatfield are shown in Figure 4.6. The first row shows what happens when this is tested on an image with an even background illumination. Using both small kernel size of 10 or a larger one of 150 results in a darker background, as shown in Figure 4.6(b) and (c). The original image shows a background of 140 (in a

range of 0 to 255), but the resulting image with a small kernel shows a background of 109. The larger kernel has a slightly uneven background ranging from 105-110. With the smaller kernel there also appears to be a bright 'halo' around the cells.

Comparing the absolute greyscale pixel value differences between the original image and the result with a kernel size of 10 shows a maximum difference of 104 shades and an average of about 32. Using the larger kernel size the maximum difference is not as large, at 76, but the average difference is about the same, at around 33.

Pseudo flatfield was also tested on our image with the manually added shading (Figure 4.6(d)). When using a kernel of size 10 there is a dark stripe left at either side of the image, which is 15 pixels wide (Figure 4.6(e)). The cells are also a lot paler than in the original image and show much lower contrast with the background.

When using a larger kernel size there is still a 15 pixel black stripe at the edge of the image, but this appears more pronounced as it graduates over the next 40 or 50 pixels. In both cases the maximum difference in shade is 140 pixels, which is where the edge pixels are zero.

As our method uses a Gaussian to filter the image, the pixel values are weighted with a higher importance given to the current pixel. Using the mean value allows the current pixel to be affected by those further away, which cause greater problems with a large kernel if the background is very uneven.

4.4.1 Results of Comparison

One of the reasons for testing on an image with an even background illumination is to test that the method makes no alterations to this. This was the case with our method, but the ImageJ results in a darker image. Despite this, it is unlikely that this would affect brightness thresholding or edge detection after running this with a larger kernel. Both of the images in Figure 4.6(b) and (c) would produce similar results to running these methods on Figure 4.6(a).

We additionally tested on an image with manually added lighting and shading so that the result could be compared with a ground truth. Again, our method showed we were able to recreate the ground truth image. The ImageJ method, however, was unable to fully remove the dark stripe at the side of the image, which would affect other segmentation operations on these images. Using a smaller kernel results in a smaller dark region, but the contrast of the cells is affected, and using a larger kernel leaves a much larger dark region, so the lighting variation problem has not been resolved.

4.5 Thresholding

In this section we will discuss the affect of applying thresholding based on brightness in order to obtain a simple segmentation of the cells. We will look at the result of thresholding both prior to and after our lighting correction technique and show what can be detected using this method.

4.5.1 Before and After

We wish to show the benefit that our lighting correction method has on brightness thresholding on our images. Figure 4.7 shows first the result of thresholding on the image shown in Figure 4.1(a), which is the image of one large cell including lamellopodia, both before and after removing the shading variations (Figure 4.7(a) and (b)). We then threshold the test image (which contains a few smaller cells) which has been manually shaded, as seen in Figure 4.5(c), again, both before and after we correct the lighting (Figure 4.7(c) and (d).

We show the results for firstly thresholding the brightest 40% of pixels and for the darkest 40% of pixels. The image ranges from 0 (black) to 1 (white), so that the thresholded image of the brightest pixels $\mathbf{T}_{BRIGHT} = [\mathbf{I} \geq 0.6]$, and the thresholded image of the darkest pixels $\mathbf{T}_{DARK} = [\mathbf{I} \leq 0.4]$. In each case those pixels which meet the criteria are

(a) Image 4.1(a) before lighting correction
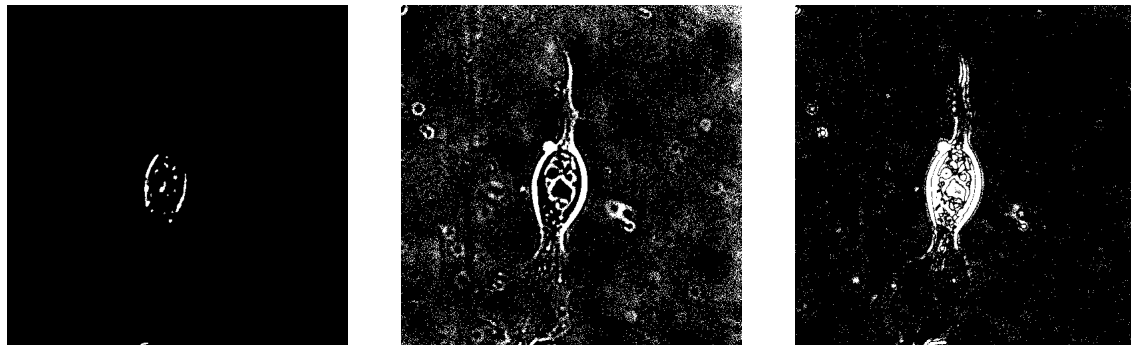
(b) Image 4.1(a) after lighting correction
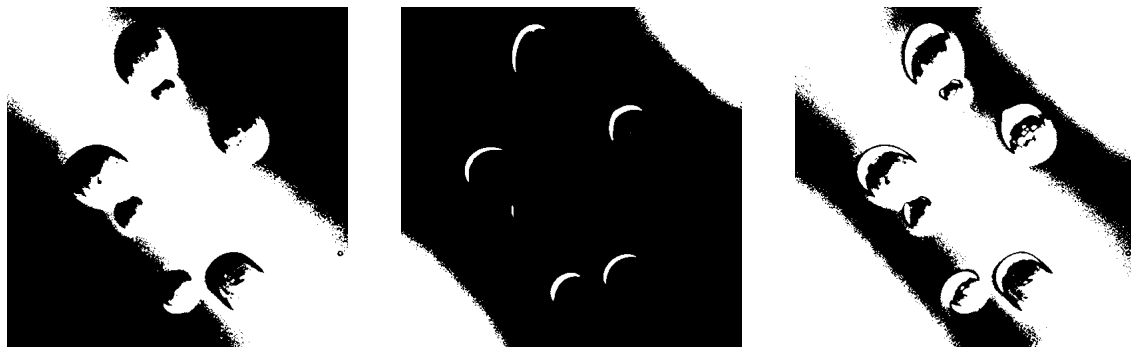
(c) Image 4.5(c) before lighting correction
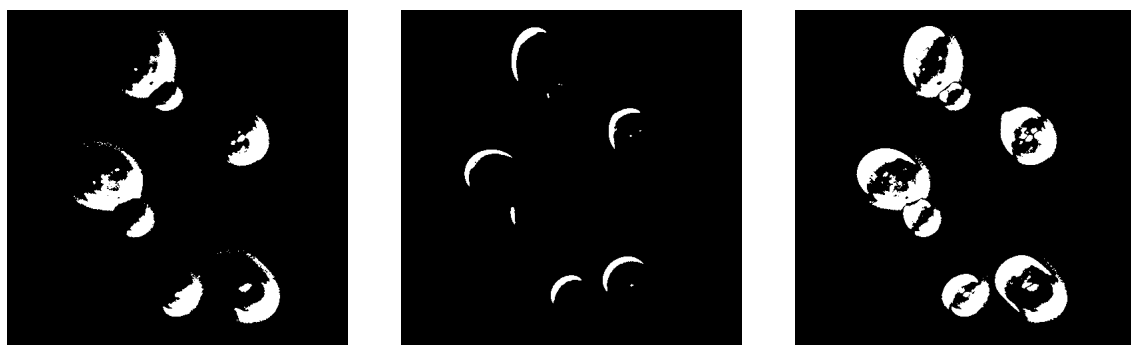
(d) Image 4.5(c) after lighting correction

(1) Brightest 40% of pixels ($\mathbf{T}_{BRIGHT}$)

(2) Darkest 40% of pixels ($\mathbf{T}_{DARK}$)

(3) Pixels outside of 10% of the mean intensity ($\mathbf{T}$)

Figure 4.7: **Images Showing Results of Thresholding on Images Before and After Lighting Correction.** We can see that with the original lighting issues brightness thresholding of the cell is not possible. After out method we are able to threshold the darker and brighter pixels to detect the cell body.

(a) Morphological operations on our real image   (b) Morphological operations on our test image
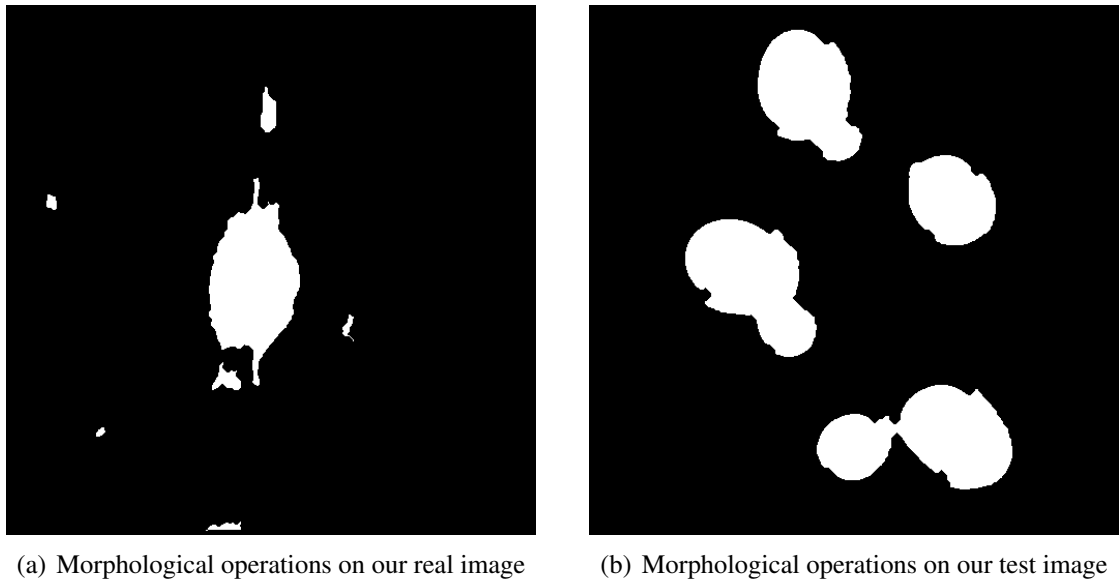
Figure 4.8: **Resulting Images After Applying Morphological Operations.** These results, showing the cell body, are achieved by applying morphological operations to the frames in Figure 4.7

shown in white and everything else is set to black.

In the final column (Figure 4.7(3)) we show the result of the thresholding used in our method. In Section 4.2 we used $\mu_\mathbf{G}$ as the pixel value for the 'background' of our image. As parts of the cell will be brighter and parts of the cell will be darker than the background we wish to find all of the pixels which are not close in value to background level. Our thresholded image $\mathbf{T} = [(\mathbf{I} > \mu_\mathbf{G} * 1.1) \vee (\mathbf{I} < \mu_\mathbf{G} * 0.9)]$.

We can clearly see from Figure 4.7(a) and (c) that when thresholding the images which have not had the shading corrected there is far too much of the background detected to allow any useful segmentation of the cells. From looking at the results after our method in Figure 4.7(b) and (d) we can see that some parts of the cell are darker than the background and other parts are brighter, so the most useful results involve using both sets of information, as in column (3).

It can be seen from the results on our test image (Figure 4.7(d)) that most of the boundary of each cell has already been detected and that it would be possible to make

a full detection using some further simple morphological operations or application of templates, which is why thresholding alone has often been used in previous work on simple shaped cells or when a full segmentation is not required. In Figure 4.8(b) we have shown the results that can be achieved by thickening the outline slightly and blurring to connect nearby components, then filling the holes on the images and thinning again.

The outcome from using thresholding alone on one of the frames from our videos (Figure 4.7(b)) shows a good result for detecting the main cell body, but very little of the lamellopodia at the bottom of the frame is detected. We have again applied the same operations as we did to the test image to connect nearby components and fill the empty holes, and the result can be seen in Figure 4.8(a). Due to the nature of DIC microscopy, large thin areas such as this will appear very similar to the background. A few parts of the edge are detected, particularly where this is ruffling up due to movement, but not enough to be able to determine the outline of the cell.

4.6   Summary

In this Chapter we have shown how the use of an efficient technique for correcting the intensity variations on DIC images can provide results which not only appear to show a more even background visually, but also confirm that the lighting is corrected with no loss of detail on a manual image and that this has no affect on a non-shaded image, proved by subtraction of the resulting images from applying our method with the original, unshaded image. The image after applying the lighting correction will be referred to as **I**. We then show that the results of this can then provide us with the information to apply a brightness thresholding which allows is to detect large parts of the cell's main body. We will refer to the binary template for the cell body as **T**.

Although this information may be enough for cells with a simple shape or those where we do not require an accurate representation of the border, in our case we are not only

looking at amorphous cells which have no standard shape but we also need to detect parts of the cell which are essential to their movement, such as lamellopodia and filopodia, which are often very fine or thin and cannot be detected in this way. We now have the basis for further work on these images, in that we have frames which are no longer affected by lighting problems (**I**) and where we can identify cell bodies (**T**). This preprocessing stage will then be built on in Chapters 5 and 6.

Chapter 5

Segmenting Lamellopodium

In this Chapter we will discuss the lamellipodia and describe some of the shape changes it undergoes during movement. We will cover why we are wishing to segment these, and some of the difficulties due to the thin, flat nature of these membranes, and due to the nature of DIC microscopy.

We will build on the result achieved in Chapter 4, which provided us with frames free from any intensity imhogeneity ($\mathbf{I}$) and a segmentation for the cell body which we can use as a basis for finding the attached lamellopodia ($\mathbf{T}$). All of the images used in this chapter are shown having had our method for correcting the shading already applied.

We will show how using edge detection can provide us with a good outline for the boundary of the majority of the cells, and compare this with leading cell segmentation software which has had the parameters optimised to provide the most accurate segmentation. We will then improve on this method using the temporal information gained from the videos, to find cells where the segmentation does not seem to have worked accurately and create an updated segmentation.

(a) Cell from blebbistatin group

(b) Cell from brenk group

(c) Cell from wildtype group
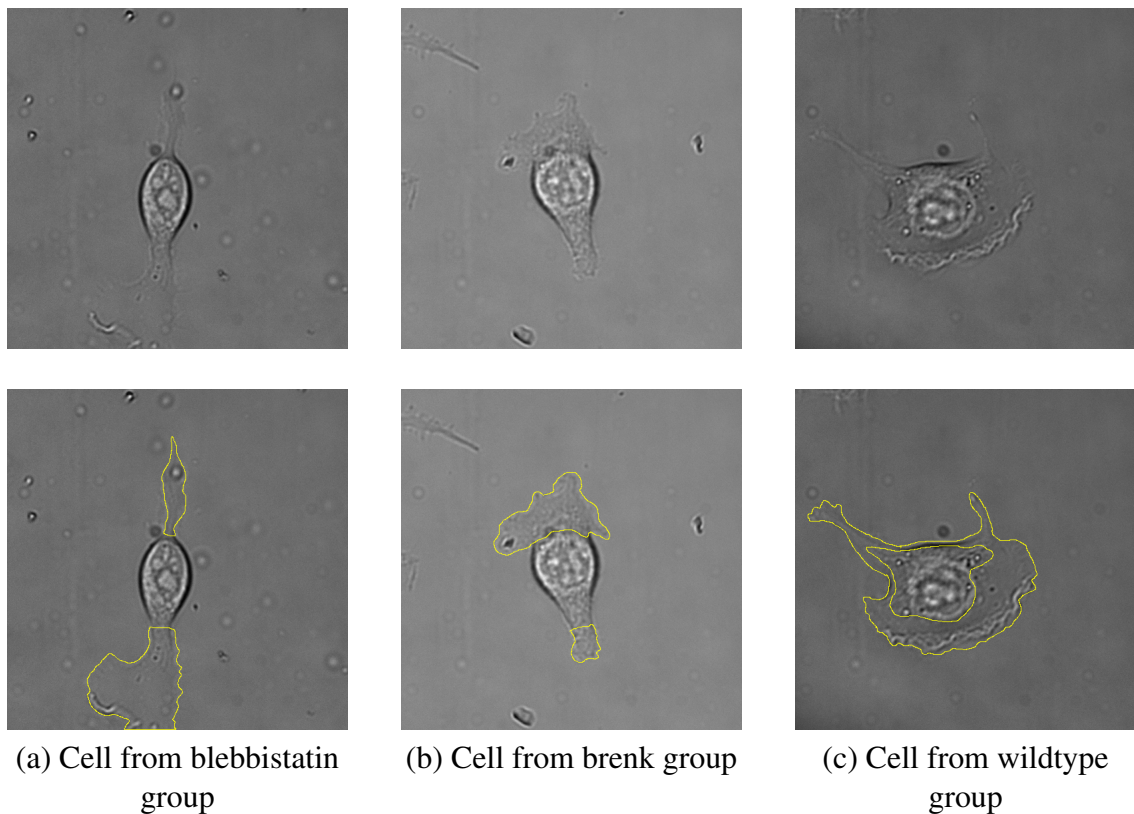
Figure 5.1: **Cells With Lamellipodia from Three Different Videos** The top row shows the frames of the video including the lamellipodia. In the bottom row these have been manually segmented from the cell body and the background.

5.1 Lamellopodia

In cells such as the myoblast cell, movement on substrate is achieved through a crawling motion. The cell extends thin pointed structures, called filopodia, which act as feelers, and larger but very thin protrusions called lamellipodia, which often spread between the filopodia. The lamellipodia stretch out and adhere to the substrate, providing anchor points for the cell to be pulled forward and pushed from the rear [5, 21, 34]. The diagrams shown in Figure 5.2 show the lamellipodia in crawling cells from the side [4] and from the top [12] of the cell. This shows how the lamellipodia attaches and detaches from the substrate as the cell moves.

At the leading edge, as the cell begins to move, the lamellipodium lifts from the sub-

(a) A Side-On Diagram Showing Lamellipodia [4]

(b) A Top-Down Diagram Showing Lamellipodia [12]
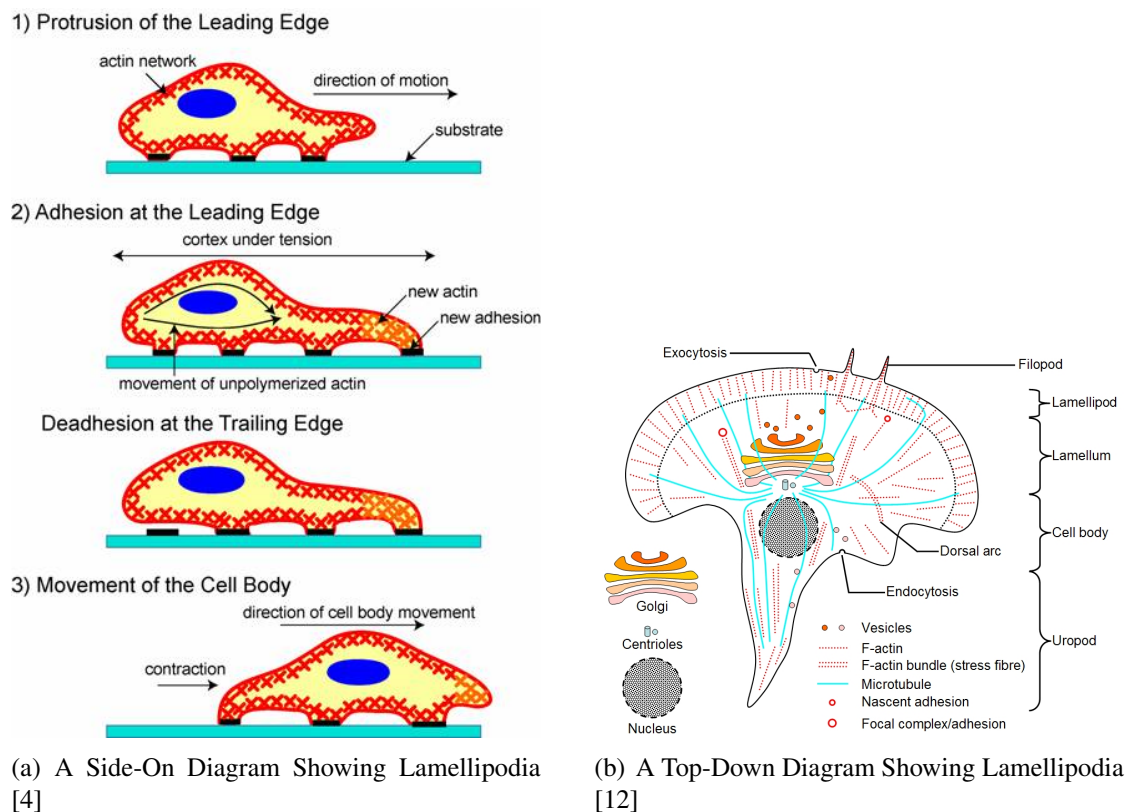
Figure 5.2: **Diagrams of Crawling Cells.** These diagrams show a crawling cell from above and from the side, showing the spreading lamellipodia at the front and rear.

strate in a wave like movement, often referred to as 'ruffling'. The ruffles appear to move back towards the cell as the edge of the lamellipodia moves forward. The ruffling can be seen in Figure 5.1. A small amount can be seen towards the bottom of the first image, where it appears brighter as it is thicker at this point. It also appears to surround the leading bottom edge of the third image, which also clearly shows the tail appearing towards the rear of the cell. As the cell is drawn forward, the rear of the cell can become a narrower 'tail' until it pulls away from the substrate [8, 34]. A diagram of the rounded forward edge and the tail is shown in Figure 5.2.

(a) A region of lamellopodia edge is higlighted
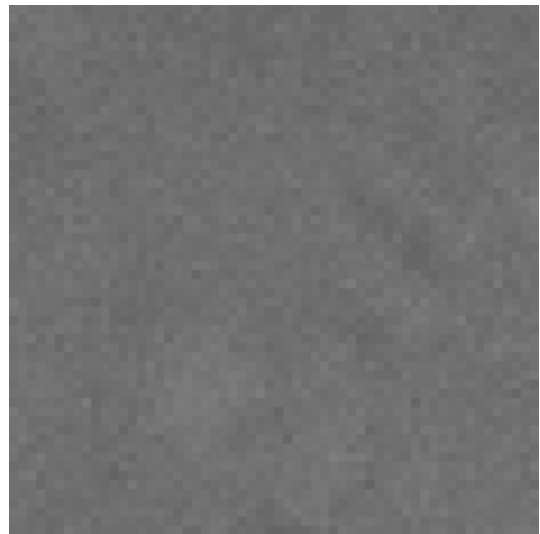
(b) A close up of the highlighted region

Figure 5.3: **A Close Up Region of Lamellipodia Edge.** The first picture shows a cell with an edge area of the lamellopodia highlighted. This region is then extracted and shown magnified. We can see from this how indistinct the edges are and that there are no visible gradients along the edge.

5.2 The Problem

Due to the nature of DIC imaging, the thinly spread membrane of the lamellipodia are often barely visible, with the texture appearing the same as the background (see Figure 5.1), except in the areas of ruffling where the lift from the substrate causes a greater thickness to be recorded. Due to this, texture analysis of these images is not able to find large parts of the lamellipodia. We can usually see this by eye as we are aware of the edges surrounding it.

Despite being able to usually 'see' the edges, in many cases we are filling in large gaps in the image as our visual systems can compensate for this. Figure 5.3 shows an example of this, in which we perceive the whole boundary but in reality a close up shows us there is not a consistent edge which is visible. The first image (Figure 5.3(a)) shows the whole cell and lamellopodia and the second (Figure 5.3(b)) shows only the highlighted region but this time magnified. The pixel values do not provide us with enough information to

be able to define where the edge is without being able to view the rest of the image.

This lack of continuous boundary is one of the big problems with many techniques which are normally used to detect edges, as there is no edge there. In some cases our visual systems cannot do this through one frame alone, but we can still perceive the edge when viewing a video due to the motion and the information from the preceding and succeeding frames.

We need to find a method which is capable of detecting the full boundary in every frame, regardless of lack of contrast or continuity of edges.

5.2.1   Biological Significance

Biologists have taken videos of moving cells using time-lapse DIC microscopy. Researchers in cell motility wish to obtain information about the cells and their movement, both in unaffected (wildtype) cells, and in those in which genes have been affected (knocked down or over expressed) or chemicals added. One particular chemical, blebbistatin, is an inhibitor which has been found to affect the cells contractile forces and has an affect on the lamellipodia and the cells' motility [24, 30, 53]. Researchers wish to know if brenk also has a similar affect on the lamellipodia, and therefore the cells' movement.

To find out more about the lamellipodia they need to be segmented separately from the cell in order to calculate information about their size and shape, and how these change over time. This work aims to segment not only the cell but also divide it into its component parts to enable this information to be calculated.

The testing will be carried out over three sets of cell videos (wildtype, blebbistatin and brenk), with 10 videos in each.

(a) Threshold 0.03   (b) Threshold 0.04   (c) Threshold 0.05

(d) Threshold 0.06   (e) Threshold 0.07   (f) Threshold 0.08
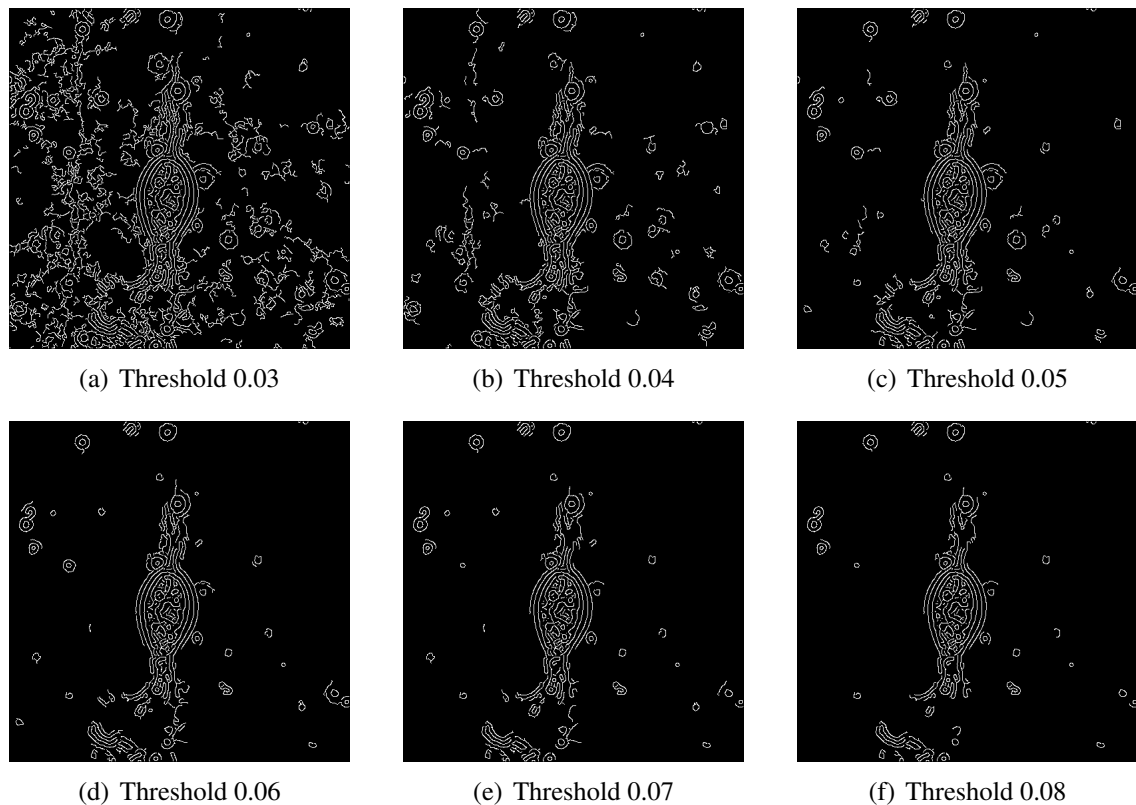
Figure 5.4: **Using Edge Detection Alone.** These images show the result of applying only edge detection to our images. They show the output from a Canny edge detection with a high threshold level set from 0.03 (a) up to 0.08 (f). It can be seen that although it will detect the edges a high degree of background is also detected.

5.3 Edge Detection

Although edge detection has previously been used to identify cells in DIC microscope images [58] this method was only suitable for circular or elliptical cells, unlike ours. It also requires constraints to be set relating to the curvature cost of the contours, whereas we are aiming for a fully automated method. From testing the most popular edge detection methods available, Canny edge detection [9] was found to provide the most accurate representation [23].

Canny edge detection uses two thresholds to identify edges. The higher threshold identifies 'strong' edges, which are those where there is a high gradient in the image intensity. It then uses a lower threshold to find those edges which do not have such a high

(a) Edge detection on blebbis-
tatin cell

(b) Edge detection on brenk

(c) Edge detection on wildtype
cell
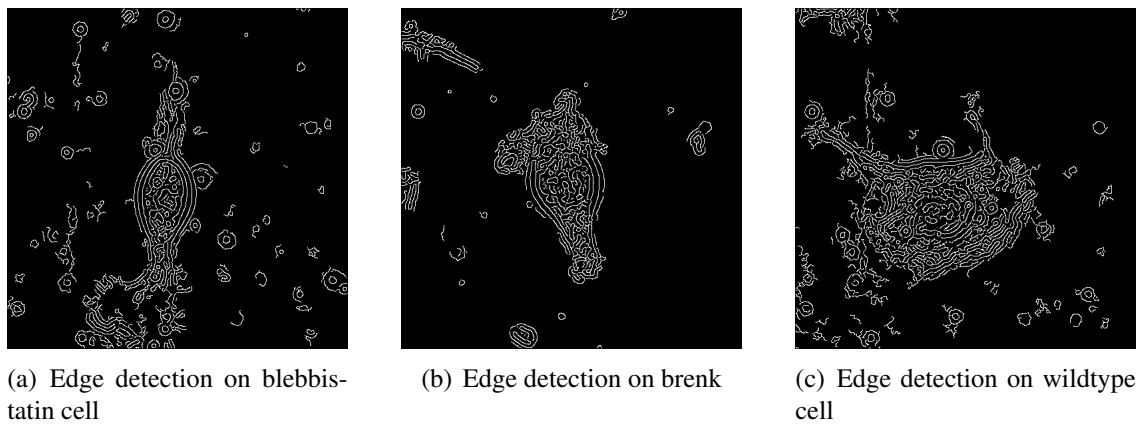
Figure 5.5: **Edge Detection Alone on Three Images.** These images show the result of running just edge detection on the images from Figure 5.1. It can be seen that although the boundary of the lamellopodia has been found, this has additionally found some areas of background.

gradient but are connected to strong edges. This is beneficial on our images where the edges of the cell cannot be easily detected due to low contrast boundaries.

Figure 5.4 shows the result of running a Canny edge detection on one of our images with a high threshold of 0.03 to 0.08 and a low threshold at 40% of each of those values. It can be seen from Figures 5.4(a) and (b) that the boundary of the cell is fully detected, but along with this so is a large area of the background. As we increase the threshold of the edge strength we decrease the background which is detected, but we also break the detection along our boundary as the faint edge of the lamellopodia is no longer being found, as can be seen in Figure 5.4(c)-(f). By the last image, with a high threshold of 0.08, only the higher contrast ruffling is being located at the bottom of the image.

From the results shown in Figure 5.4, we would be able to detect the lamellopodia but would be unable to do so without also detecting areas of the background which would need to be manually removed, or would involve parameters setting individually to enable only the cell to be segmented. Figure 5.5 shows the results of running edge detection on the three images shown above in Figure 5.1. In each case the cell boundary is found, but so are differing degrees of non-cellular material from the background of the image. We

(a) Original Image **I**   (b) Edge Map **E**   (c) Edges and Thresholding **E** ∨ **T**

(d) After Filtering   (e) Binary Image   (f) Filled Image
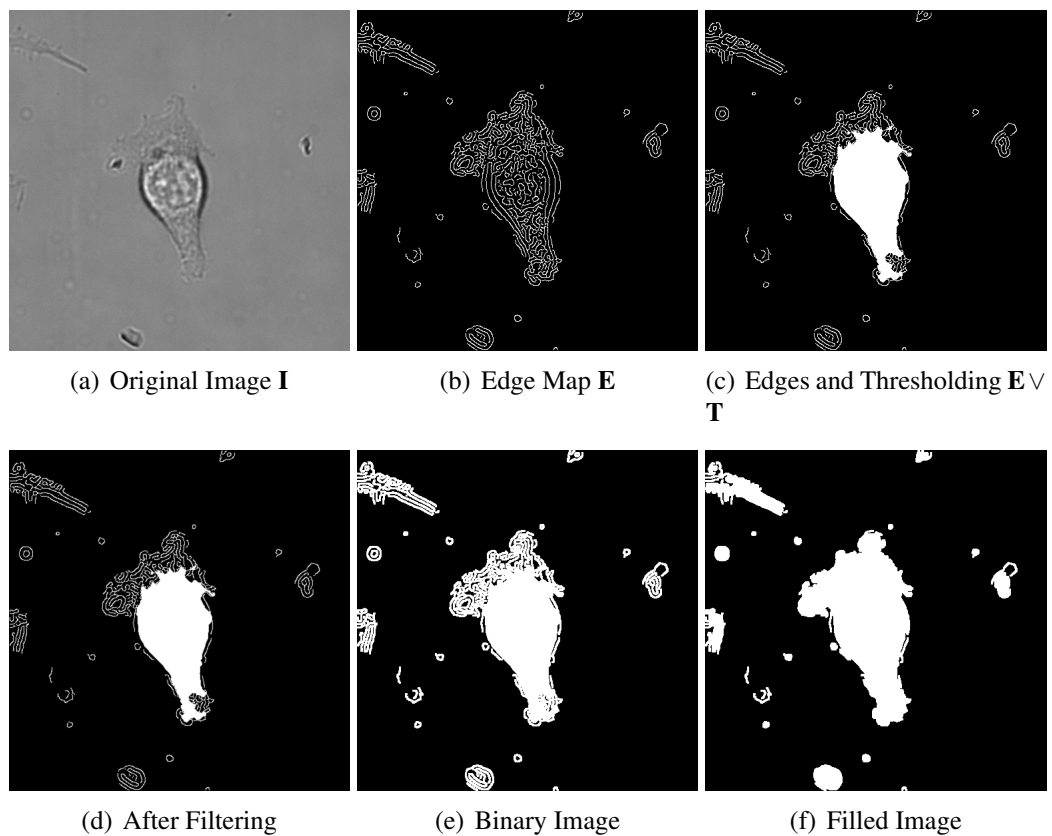
Figure 5.6: **Stages of Morphological Operations.** We show the original images then the stages followed to connect and fill the template of the cell.

will refer to the resulting binary edge map, in which pixels are stored as a 1 if they are part of a detected edge, as **E**. However, we are already aware of the cell's location from the thresholding applied in Section 4.5 so we are able to use this information to find only the edges which can be connected to the cell's body.

5.4   Combining Thresholding and Edge Detection

As mentioned in Section 5.3, we can find the cell boundary using edge detection but this also provides us with a degree of background inaccurately being segmented also. We could try to optimise the edge detection further, but no method was found which could do this in an automated manner as the parameters would have to be changed for every frame.

As we are already aware of the location of the cell body from Section 4.5 we can use this to find only those edges which are connected to the cell's body.
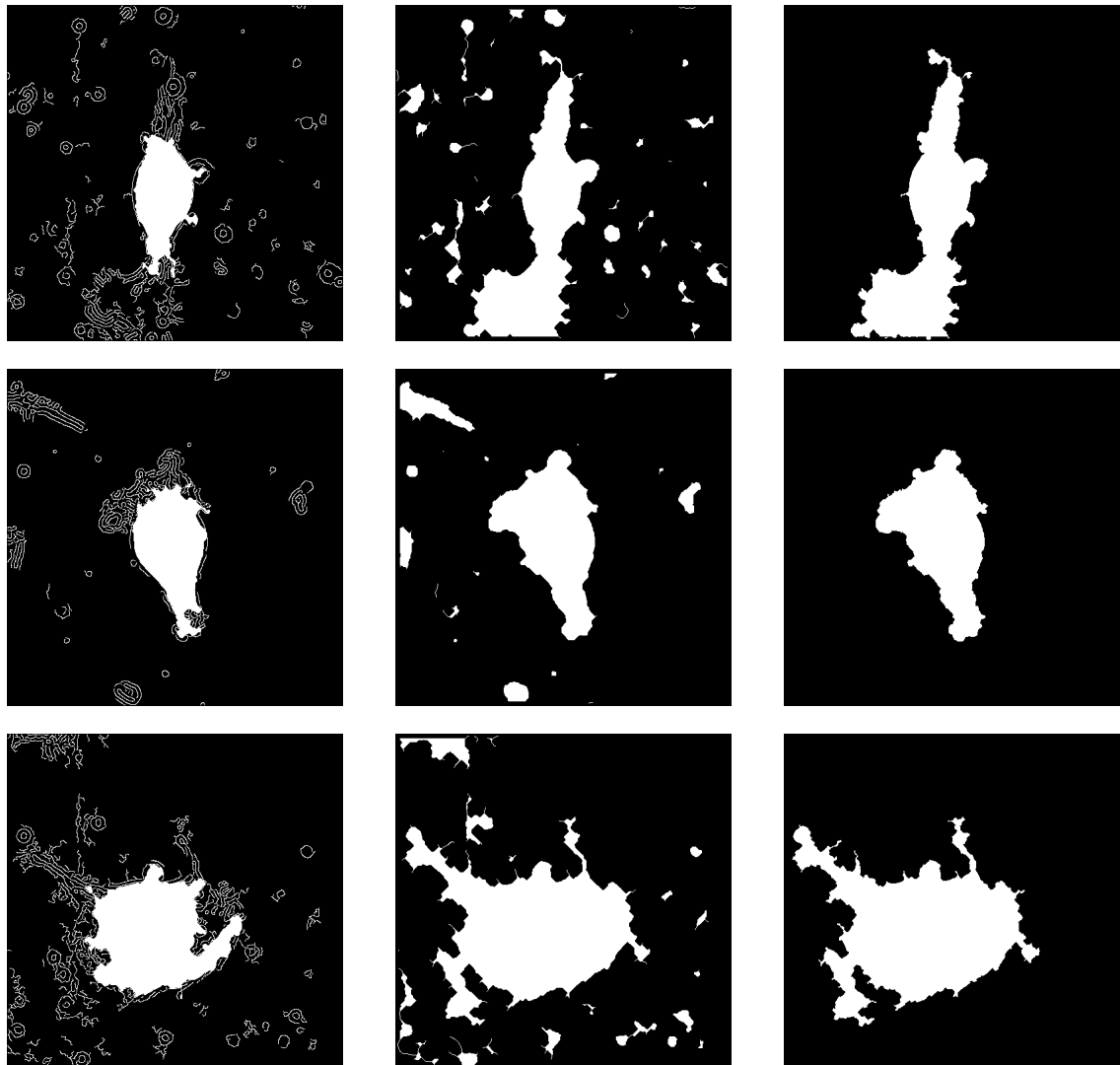
| (a) The combined thresholded body from Section 4.5 and edge detection from Section 5.3 | (b) After morphological operations to connect and fill the image. | (c) After re-application of the edge detection to smooth and correct the edges. |

Figure 5.7: **Combining Edge Detection And Thresholding.** These images show the process of combining the edge detection and thresholding results and using morphological operations to find the outline of the cell. It can be seen the the first two cells have far more accurate shapes than the third (wildtype) cell.

We again use morphological operations to connect the edges to the cell and fill the
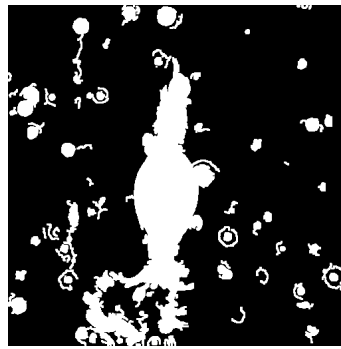
Figure 5.8: **A** Cell Which Has Not Been Filled. As the detected edges do not completely surround the lamellopodia the bottom region has not been filled

result. In Figure 5.6(a) we show the original image **I** and in (b) we show the binary edge map **E**. We then combine the result of our thresholding and edge detection so that Figure 5.6(c) shows $\mathbf{E} \vee \mathbf{T}$, in that any pixel which is a 1 in either of our binary images is a 1 in our combined image.

We then filter the image using a Gaussian with a σ of 0.5 using a small (3×3) window, the result of which can be seen in Figure 5.6(d). At this stage in the process the image is no longer a binary image, which is why it is difficult to detect the effect in this image, as some of the pixels value are very low and almost indistinguishable from black. In Figure 5.6(e) the image is then converted back to binary, by treating all pixels above zero as one. This stage has connected those pixels which are up to three pixels apart.

To create a solid template we apply a fill operation on the 'holes' in the image. For a binary image this would mean that any black pixel, or block of black pixels, that is entirely surrounded by white pixels would be set to white. In a grey scale image an area completely surrounded by a brighter area would be filled to the intensity of the brighter area. The fill operation only takes place if the region is completely surrounded, in our case using and 8-connected neighbourhood. The result of this can be seen in Figure 5.6(f).

If there are not strong enough edges around the lamellopodia for them to connect entirely then the region will not be filled. An example of this can be seen in Figure 5.8. This leads to problems, as shown in Figure 5.10 and discussed in Section 5.5.2.

As it can be seen from 5.6(e) and (f), the image is now thicker than previously due to the previous operations. To remove the additional pixels we 'thin' the object by removing two layers of pixels from around the boundary.

Figure 5.7 shows the comparison of these step on cells from three different videos. Figure 5.7(a) firstly shows the edge detection and thresholding, $\mathbf{E} \vee \mathbf{T}$. We then show the result of the operations from Figure 5.6 and the thinning in (b). At this stage there are still some non-cellular objects left in the scene. As we are aware that the connected cell must be at least the same size as the cell bodies alone we can use information from Section 4.5 to remove these areas, as we know the size of the smallest component in $\mathbf{T}$ we can remove anything smaller than this. Due to the thickening and thinning operations, the resulting cell outline sometimes misses some of the detail which is shown in the edge detection, so we add this again to the final image and refill to include the small outline changes, as can be seen in Figure 5.7(c). We will refer to this binary template for the segmentation of a cell as $\mathbf{C}$.

(a)   (b)   (c)
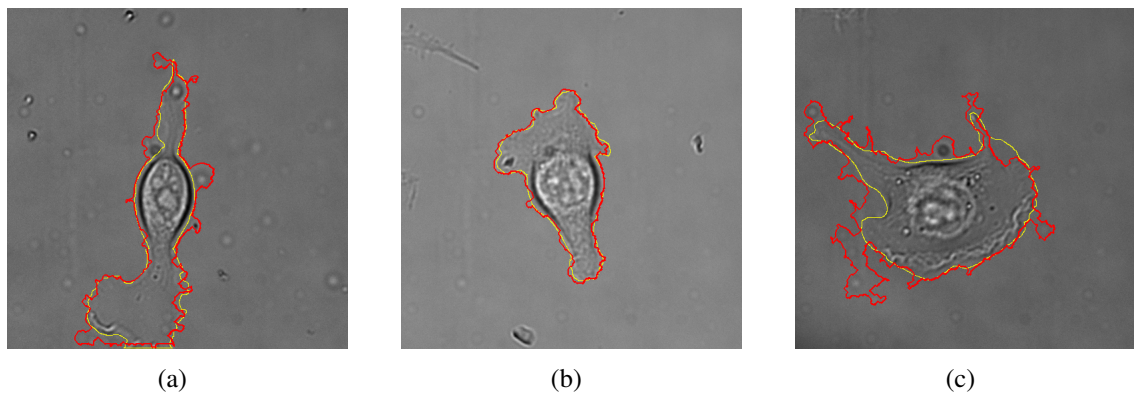
Figure 5.9: **The Manual and Automatically Created Boundaries.** We have shown the manually drawn boundary of each cell in yellow then drawn our automatically created boundary in red. In the first two images the boundary is quite a good representation but the third image shows an area where the correct edge could not be found and some of the background has been added to the cell.

In Figure 5.9 we show the results of our automatically created boundaries from $\mathbf{C}$ on the cell images $\mathbf{I}$ in red, with the manually marked cell showing in yellow. It can be

seen from the first two images that the boundary is quite close to the actual cell. Figure 5.9(a) shows some areas where small items which are touching the cell have also become attached, whereas Figure 5.9(b) shows quite an accurate representation of the whole cell. Figure 5.9(c) is an example of where a bigger problem has occurred as the edge appears to have been completely lost and the boundary has started 'leaking' out into to the image.

5.5 Initial Results

At this stage we wished to find out the current accuracy of our implementation, which was done by manually marking up every 10th frame from each video, over our three sets of blebbistatin, brenk and wildtype videos, and comparing the segmented cell with our automatically segmented cell. We used both the DICE and Jaccard measures of similarity to assess our results.

5.5.1 DICE and Jaccard

Both DICE and Jaccard are used to measure the similarity of two samples. In terms of segmentation, we are comparing the binary result of the segmentation, such as can be seen in Figure 5.7(c), with a binary result of a manually segmented image.

The DICE measure, shown in Equation 5.1, compares the size of the intersection with the sum of each, by taking double the size of the intersection of the automated segmentation and the manual segmentation and dividing this by with the size of the sum of pixels in each segmentation. The Jaccard measure, shown in Equation 5.2, again looks at the intersection of the segmentations but this time compares with the size of the union of the two.

Both measures give a score between zero and one, with a result of one meaning that the two binary images are the same, as we are wishing both segmentations to overlap completely.

$$\text{DICE}(A,B) = \frac{2|A \cap B|}{|A| + |B|} \tag{5.1}$$

$$\text{Jaccard}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{5.2}$$

5.5.2 Comparisons

Although the results for this system were obtained using a fully automated system, no other automated system could be found for comparison of results that was able to detect and segment the cells. The segmentation was instead compared to results using Cell-Track [43], which is popularly used for motility analysis. We manually adjusted the parameters for each video, including the initial edge threshold, the edge linking threshold, the dilations and erosions, and the size of the cells. This was done until CellTrack showed the closest possible result to the cell boundary, and involved different values for every video. We then performed the similarity measures against our manually segmented cells, in the same way as we did with our automated results.

Table 5.1: Comparison Of DICE Results (% accuracy).

	Wildtype	Blebb.	Brenk
Our Segmentation	**89.75**	**86.64**	89.33
CellTrack	88.81	85.52	**90.89**

Table 5.2: Comparison Of Jaccard Results (% accuracy).

	Wildtype	Blebb.	Brenk
Our Segmentation	**82.25**	**77.43**	81.56
CellTrack	80.20	75.49	**83.44**

Tables 5.1 and 5.2 show the results for our method compared to the cell track segmentation as an average over each set of videos, for both of the similarity measures. The most accurate results are shown in bold. It can be seen that, even at this stage, for both the wildtype and blebbistatin segmentations we receive better results than for the best available

option which necessitated manual parameter setting (best results shown in bold font).

| (a) | (b) | (c) |
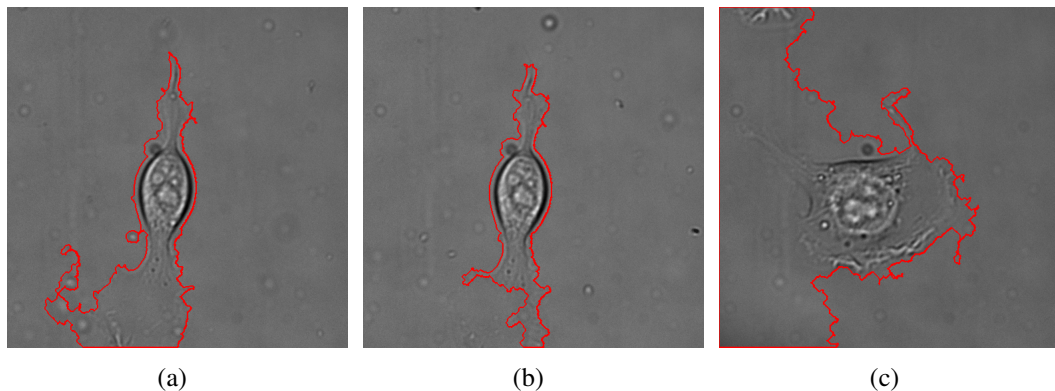
Figure 5.10: **Examples of Problem Segmentations.** These images show some examples of bad segmentations. The first two have not fully connected the lamellopodia edges. The third has a very bad segmentation including most of the frame.

Despite receiving very good results over all, there are still some individual frames that do not appear well segmented. The particular problem frames tended to receive poor results for both our automated method and using CellTrack [43]. In Figure 5.10(a) the boundary detection has not found a thin part of the edge and has started to spread into the image. In Figure 5.10(b) it has again failed to detect the edge, but this time it has completely missed segmenting part of the lamellopodia as it has found a stronger edge going across it. In these cases it may be possible to try to achieve a better segmentation by fine-tuning the parameters we have used for the edge detection, but this would need doing on an individual or per video basis so would not be suitable for our automated system.

In Figure 5.10(c) we have a completely failed segmentation for half of the image. The right half of the cell has been quite roughly segmented but the left has drifted off to the boundary of the image. Again, we do not wish to change our parameters on a frame by frame basis. From viewing the results of the segmentation on each video, it does not appear that we receive poor results over any video as a whole. Although there can be consecutive frames showing a problem, the majority of the frames show a very close

segmentation to the boundary of the cell. We wish to build on the information provided by this method to increase the accuracy of our segmentation.

5.6   Incorporating Movement and Time

The resulting segmentation for the thresholding and edge detection method was very close to the manual segmentation for most frames (Section 5.5), but in some frames the lamellipodia were not entirely detected, or background was incorrectly detected, examples of which can be seen in Figure 5.10. In these cases it was often also extremely difficult or even impossible to detect these areas by eye in the individual frame, but the location could be inferred from the movement in the surrounding frames. We took to following steps, which will be explained in more detail, along with the images in Figure 5.11:

Step 1 - Compare the cell size with the cell in the next frame. If more than 10% difference move to step 2, otherwise store the frame and move to the next frame.

Step 2 - Take the intersection of the previous and subsequent frame.

Step 3 - Apply the edge detection from the current frame.

Step 4 - Update with the new cell segmentation and move to the next frame from Step 1.

Step 5 - If at final frame, check for halting conditions. If not met, move back to Step 1, frame 1.

Step 1

To detect these cases (of incomplete segmentation) the cell size was compared over consecutive frames, with those frames showing a significant increase or decrease being the most likely to not be entirely segmented. We denote a cell segmentation at time t as $\mathbf{C}(t)$. At each time-step, we compare the area of $\mathbf{C}(t)$ with that of $\mathbf{C}(t+1)$. If the areas has

(a) Original(1)   (b) Original(2)   (c) Original(3)   (d) Original(4)   (e) Original(5)

(f) $\mathbf{E}(1)$ (g) $\mathbf{E}(2)$ (h) $\mathbf{E}(3)$ (i) $\mathbf{E}(4)$ (j) $\mathbf{E}(5)$

(k) $\mathbf{C}(1)$ Save   (l) $\mathbf{C}(2)$ Update   (m) $\mathbf{C}(3)$ Update   (n) $\mathbf{C}(4)$ Update   (o) $\mathbf{C}(5)$ Save

(p) $\mathbf{C}(1) \cap \mathbf{C}(3)$ (q) $\mathbf{C}(2) \cap \mathbf{C}(4)$ (r) $\mathbf{C}(3) \cap \mathbf{C}(5)$

(s) Reapply Edges $\mathbf{C}'(2)$ (t) Reapply Edges $\mathbf{C}'(3)$ (u) Reapply Edges $\mathbf{C}'(4)$
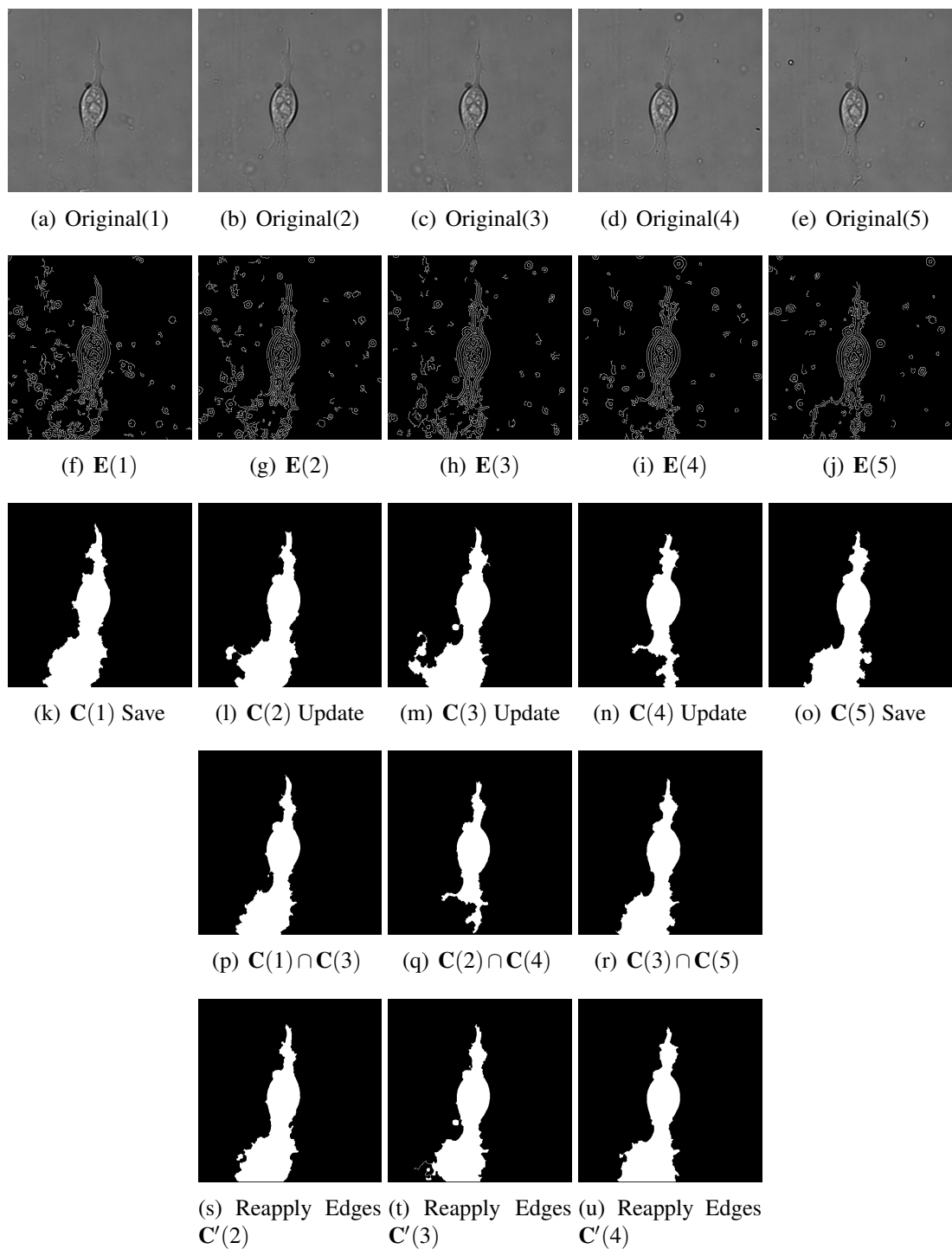
Figure 5.11: **Examples of Segmentations and Intersections.** We show five consecutive frames along with their edge maps and segmentations. Three of these differ in size from the next and would require updating. We then show the intersection of the previous and subsequent frame and the resulting segmentation after reapplying the edges.

changed by at least 10% we move on to step 2. If not, we store that cell and move on to the next frame. In Figure 5.11, the third row shows our cell segmentations. It can be seen that $\mathbf{C}(2)$, $\mathbf{C}(3)$ and $\mathbf{C}(4)$ show size differences from the next frame. These would be updated in Step 2.

Step 2

To generate the basis of a new segmentation image we take the intersection of previous and subsequent frames, which we will show as $\mathbf{C}(t-1) \wedge \mathbf{C}(t+1)$. In this case we are taking the intersection of these binary images to be the overlap of the white pixels, so that is where $\mathbf{C}(t-1) + \mathbf{C}(t+1) = 2$. If we are at the first frame we use the frame itself, $\mathbf{C}(t) \wedge \mathbf{C}(t+1)$.

Some examples of these intersection are shown in Figure 5.11, row 4. It can be seen here that we already have an improved segmentation for $\mathbf{C}(4)$, but $\mathbf{C}(3)$ does not appear as good. This will be resolved in Step 3.

Step 3

As stated in section 5.3, we have a set of edges $\mathbf{E}$, which we denote for each time step as $\mathbf{E}(t)$ generated from $\mathbf{I}(t)$. We apply the edge map to the intersection from step 2, so that at time (t) we take $(\mathbf{C}(t-1) \wedge \mathbf{C}(t+1)) \vee \mathbf{E}(t)$. The result is a binary image consisting of the intersection from the previous and subsequent frame, along with the edges from the present frame set to white and the remainder as black.

For frame $\mathbf{C}(4)$ from Figure 5.11(n) we show this process in more detail in Figure 5.12. It can be seen that the segmentation in (b) does not include a large part of the lamellopodia. When we look at the edges ($\mathbf{E}$) it shows that they do not fully connect, and therefore will not be filled.

The intersection image is shown in Figure 5.12(d). This is already an improvement on the previous segmentation. In (e) we show this along with the edges for this image.

We then use the morphological operations described in Section 5.4 to connect the edges, fill the shape, and keep only those parts which are connected to the cell. We show our updated cell segmentation, $\mathbf{C}'$ in Figure 5.12(f).

In step 2 we mentioned the segmentation of $\mathbf{C}(3)$ from Figure 5.11. Taking the intersection of the previous frame with a good segmentation and the subsequent frame with a bad segmentation appears to have given us a worse result as shown in Figure 5.11(q). However, it can be seen that the edges from $\mathbf{E}(3)$ do provide a good representation of the boundary, so that the end result when these are reapplied is an improved segmentation.

Step 4

At this stage we update our segmentation for $\mathbf{C}(t)$ with $\mathbf{C}'(t)$. This process runs on a frame by frame basis, so that each segmentation is updated before moving on to the next frame, therefore when making any changes to cell $\mathbf{C}(t+1)$ it will be using the updated segmentation for cell $\mathbf{C}(t)$.

If we are not at the final frame of the video we move to the next frame and return to Step 1.

Step 5

When we reach the end of the video, to check if the process needs to be repeated we firstly check the size differences of the cells. If no consecutive frames differ by more than 10% then the process stops. We also store a list of sizes for each run and the process stops if no further changes to the size are being made. If these criteria have not been met then the process is repeated from Step 1.

A final stopping criteria was implemented in that the number of runs should be no more than the number of frames. This was a 'fail safe' implementation to prevent the system becoming stuck in a loop where if could alternate between segmentations over two or three cycles, although in practice this did not occur.

(a)   (b) $\mathbf{C}$ (c) $\mathbf{E}$

(d) $\mathbf{C}(t-1) \wedge \mathbf{C}(t+1)$ (e) $(\mathbf{C}(t-1) \wedge \mathbf{C}(t+1)) \vee$ (f) $(\mathbf{C}'(t)$
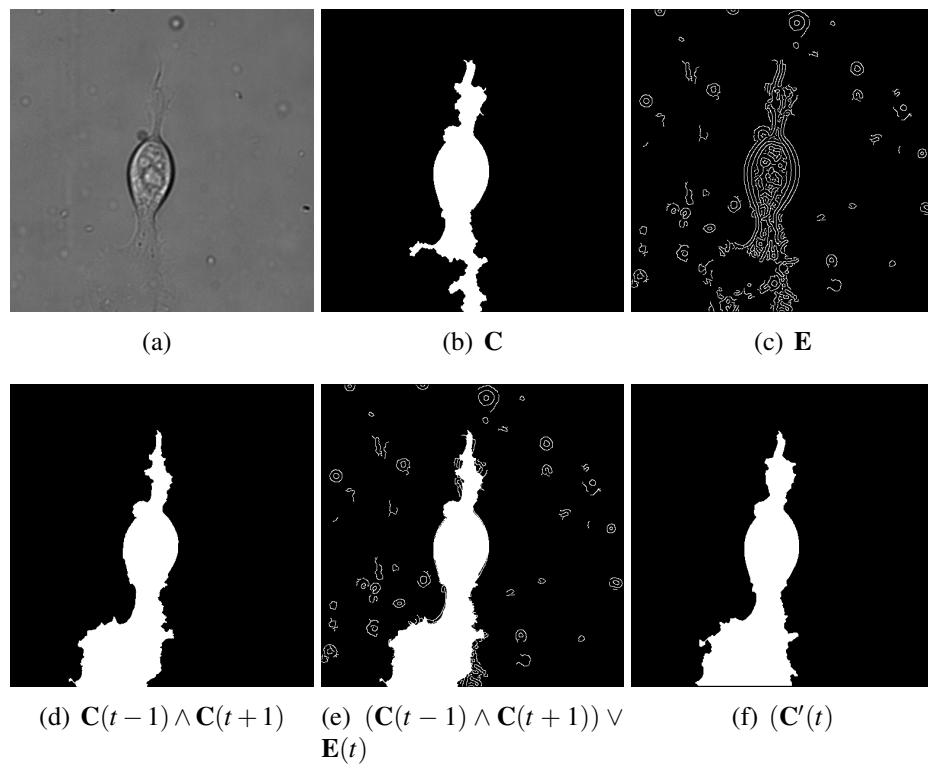 $\mathbf{E}(t)$

Figure 5.12: **Steps to Update a Segmentation.** This figure shows the poor segmentation from Figure 5.11, $\mathbf{C}(4)$. The edges show why the lamellopodia was not segmented. When taking the intersection of the previous and subsequent frame and reapplying the edges we receive a much closer segmentation.

Benefits

The advantage of this iterative framework becomes evident where we have two or more consecutive erroneous segmentations. Take as an example the situation where $\mathbf{C}(t)$ and $\mathbf{C}(t+1)$ are incorrect; when taking the intersection of $\mathbf{C}(t-1)$ and $\mathbf{C}(t+1)$, we still obtain an incomplete representation of the cell. As such the initial pass of the method will not create a full segmentation throughout the video. However, if we repeat the full process as described above, segmentation improves at each pass until we converge on a more complete solution. In general, the system would halt after two or three cycles due to not being able find any further cell size difference problem.

5.7 Results

<div align="center">(a)   (b)   (c)</div>
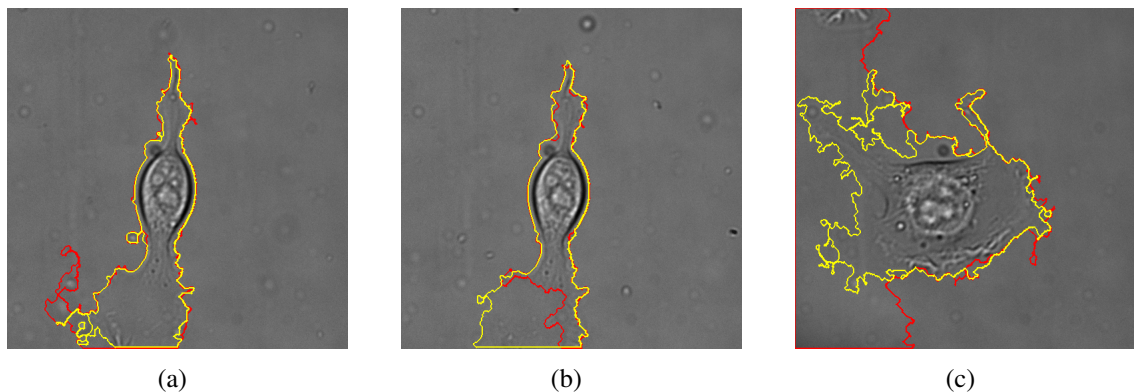
Figure 5.13: **Examples of Improved Segmentations.** These images show the initial segmentation in red and the segmentation after our new method in yellow. Pictures (a) and (b) now show a quite accurate representation of the cell boundary. Although image (c) is still showing problems it is a vast improvement on the Figure 5.10(c).

After running our method, which looks at large changes in cell size and then uses the temporal information from the surrounding frames of problem boundaries to improve the segmentation, we can see visually that improvements have been made, as shown in Figure 5.13. We have highlighted the segmentation from the first method in red and the updated

Table 5.3: Comparison Of DICE Results (% accuracy).

	Wildtype	Blebb.	Brenk
Old Segmentation	89.75	86.64	89.33
CellTrack	88.81	85.52	90.89
Updated Segmentation	**92.61**	**87.03**	**92.67**

Table 5.4: Comparison Of Jaccard Results (% accuracy).

	Wildtype	Blebb.	Brenk
Old Segmentation	82.25	77.43	81.56
CellTrack	80.20	75.49	83.44
Updated Segmentation	**86.50**	**77.77**	**86.59**

segmentation in yellow. Figure 5.13(c) is showing a massive improvement on the original segmentation but is still not close to the actual boundary at quite a few parts of the cell. Some ideas to correct this further are suggested in Section 5.8.

(a)   (b)
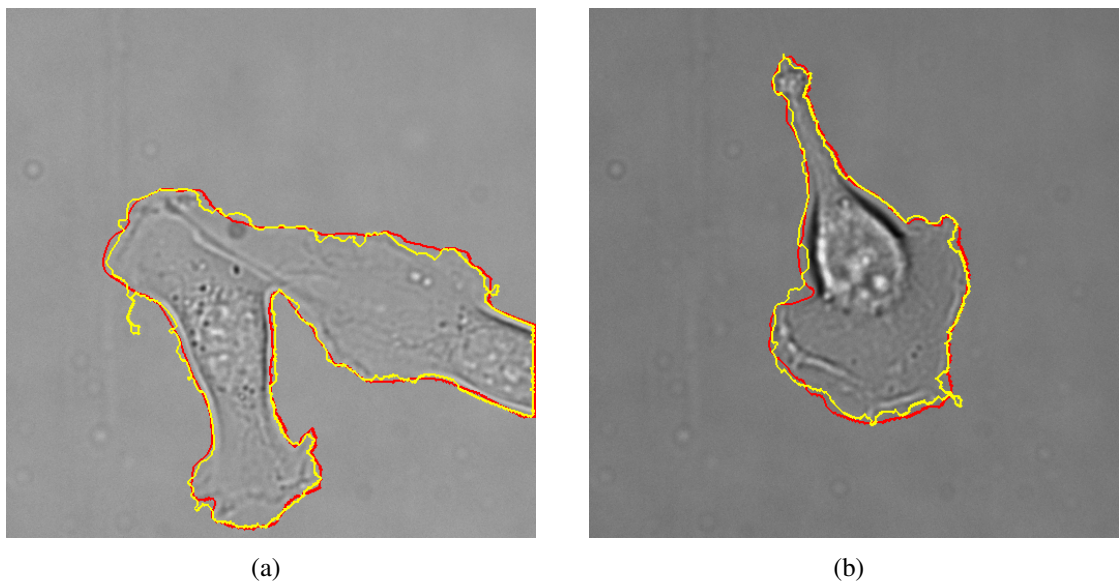
Figure 5.14: **Examples of Good Segmentation.** These images show some examples of a good segmentation. The manual segmentation is shown in red and the automated segmentation from our iterative improvement is shown in yellow.

Tables 5.3 and 5.4 show the updated accuracy per group after using our iterative correction method, again using the DICE and Jaccard similarity measures to compare the

results with a manual segmentation. In every case the result is an improvement on our previous method, but is now also more accurate that CellTrack in every case. Obviously, we do not want just high accuracy as an average but on a frame by frame basis. Table 5.5 shows the percentage of frames in the test set which fall into each accuracy rating. Overall, 74% of frames have an accuracy of over 90% when using the DICE similarity score, and 95% of frames are more than 80% accurate. Two examples of a good segmentation are shown in Figure 5.14. The manual segmentation is shown in red and the segmentation after running our iterative method is shown in yellow. Both images show a boundary which represents the cell shape.

The frames with the lower accuracy (less than 80%) all occurred across three videos. Some examples are shown in Figure 5.15. In both Figures 5.15(a) and (b) it can be seen when looking closely at the image that the boundary is very hard to detect. Figures (c) and (d) show the manual segmentation in red, although this may also have some inaccuracies to the difficulty described in seeing the boundary and therefore manually determining the boundary. Figure 5.15(c) shows that when the correct boundary has not been found the edge has moved away into the background. In Figure 5.15(d), however, the opposite occurred and only the inner edges of the cell have been detected so large parts of the lamellopodia are missing.

In the worst segmentation cases (those lower than 40%) the entire frames was detected as being part of the cell. This occurred across a few frames in one video. The problem has occurred because the edge detection threshold is not suitable for this frame and would work better if set higher so less of the background was detected. Some suggestions to correct this are discussed in Section 5.8.

(a)

(b)

(c)

(d)
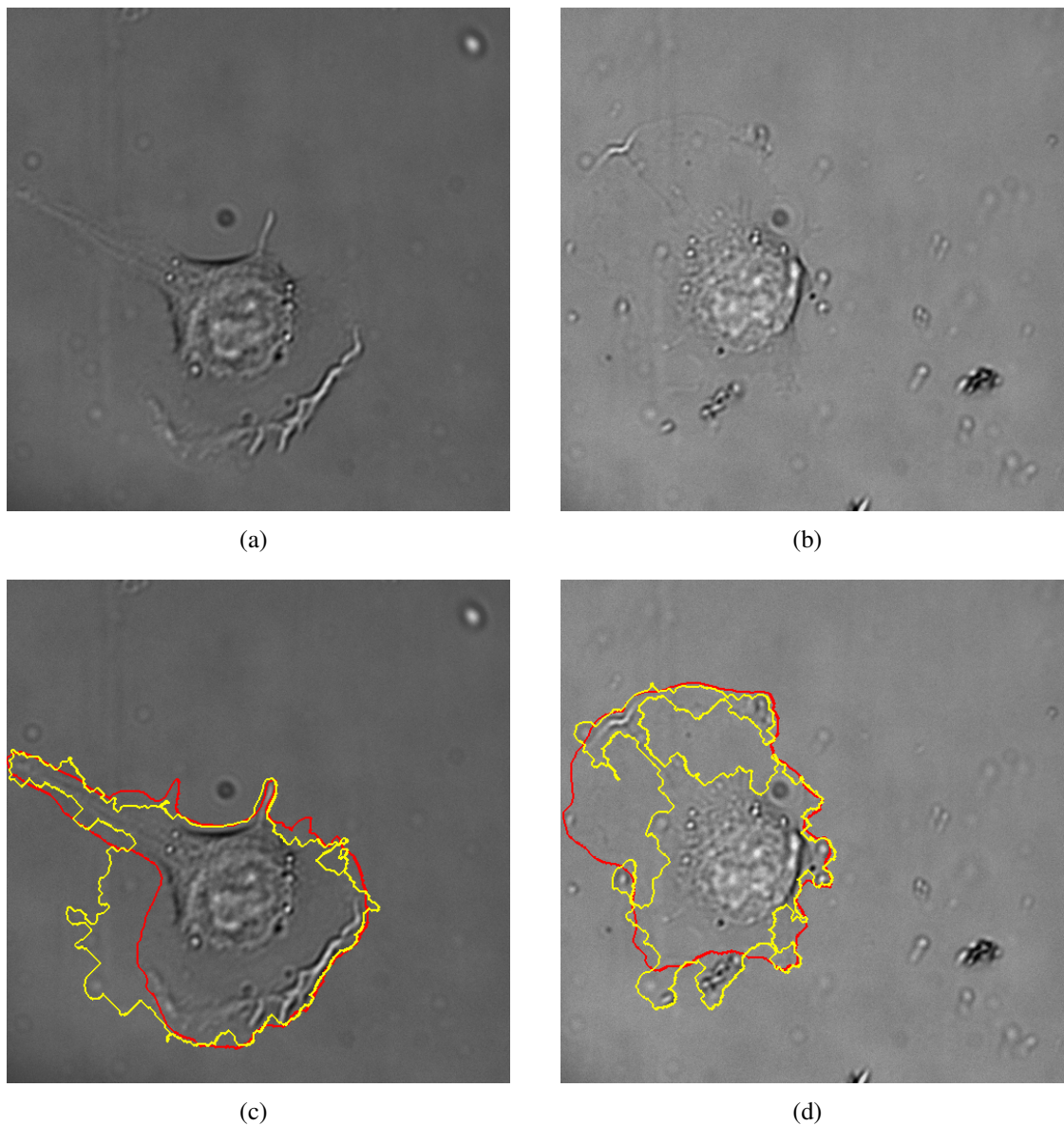
Figure 5.15: **Examples of bad Segmentation.** These images show some examples of bad segmentations, with the manual segmentation again shown in red and the automated segmentation shown in yellow. Above these we have shown the original images so that the poor boundary can be seen.

5.8   Further Work On Lamellopodia Segmentation

In Section 5.7 we saw that in a few cases the edge detection was so poor that the entire frame was detected. To counter this a further check was implemented to compare the size of the detected 'cell' to the size of the frame. If the size was more than 95% of the frame

Table 5.5: Accuracy on Percentage of Test Frames.

Accuracy %	Frames %
90 - 100	74
80 - 90	21
70 - 80	2.5
60 - 70	0
50 - 60	0
40 - 50	0
30 - 40	1.2
20 - 30	1.2
10 - 20	0
0 - 10	0

it was assumed that the edge detection was incorrect and could not be used to complete the boundary of the cell. Instead, we took only the intersection of $\mathbf{C}(t-1)$ and $\mathbf{C}(t+1)$ to create the new segmentation image.

If the problem is occurring in consecutive frames then those between problem frames would not be fixed on the first run of this, but would improve with further iterations. Some examples of this are shown in Figure 5.16, but it can be seen from (a) that the problem has not been resolved. This frame is between other bad frames where the segmentation is not good but is not covering the full screen. In the second example (Figure 5.16(b)) we can see that we now have a very close segmentation, which has 94% DICE similarity measure, and brings the total of our test frames with over 80% accuracy to 96%. Unfortunately, this is only a 'best guess' approach and could be improved open by taking more information about the location of the cell's boundary over a greater number of frames. This will be discussed in more detail in Chapter 7.3, Future work.

A better solution may be to try to improve on the edge detection that we are using, by altering the threshold for this based on whether the cell is dramatically larger or smaller than the previous frame. This will again be discussed in our Future work Chapter (7.3)

(a)   (b)
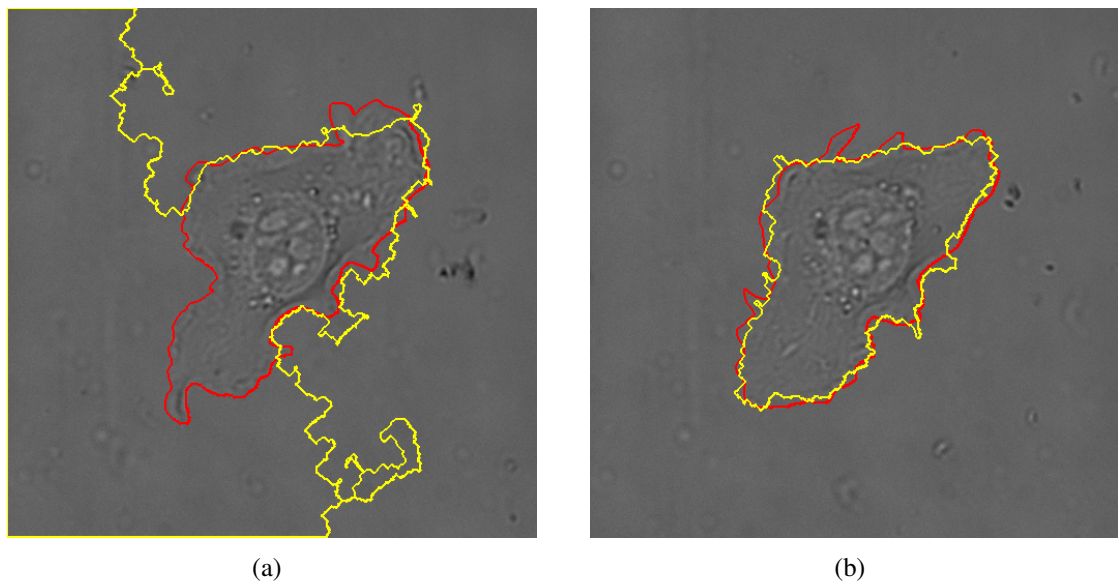
Figure 5.16: **Correcting For Problem Edge Detection.** These images show some how those frames where the entire image is selected for segmentation after running the method to take only the intersection and not reapply the edges. It can be seen in (a) that there is still a large problem due to not having a good segmentation in the surrounding frames, whereas the second frame has increased to 94% accuracy (DICE).

5.9 Summary

We have shown that applying our edge detection method to our previously found cell bodies (**T**)), from Chapter 4, provides a good representation of the cell boundary in most cases (**C**). We then improved on this using an iterative method taking the temporal information from the videos. This was used as often the outlines are not even visible to the human eye unless the cell is seen moving in the video. This system is fully automated as all the parameters are set internally based on the information it takes from the video, and all adjustments are made based on our segmentation from other frames. Despite this, we receive better results than one of the most popular programs for cell segmentation in which all parameters were manually fine tuned for optimal segmentation for every video. We could not compare with any other automated system as there is none available to segment cells with such variability in shape and with thin membranes.

This method works regardless of the number and size of cells, although it may fill gaps in between clustered cells. This could be solved for regularly shaped cells as templates could be provided, but would be unlikely in cells such as these.

5.10 Other Methods

As mentioned previously, a number of other methods were tested for the segmentation of the lamellipodia. None of the methods produced any improvement on our system. We will discuss some of the alternative ideas below along with why these did not prove suitable for our automated segmentation system. These will include thresholding alone, active contours and texture analysis.

In Section 5.3 we showed the results of using a Canny edge detection with varying thresholds. There was no level which could detect the lamellopodia without also detecting a high degree of the background, and the degree of segmentation for the same threshold differed for each image. Without any prior knowledge of cell size or shape it is not possible to use the edge detection alone to segment the cell.

5.10.1 Brightness Thresholding Alone

Thresholding has been a popular method for segmenting cells, particularly those filmed using GFP, as discussed in Section 2.6.1. It has also been used on some cells filmed using DIC, as discussed in Section 3.2.2, such as work by [57] which involved manually selecting the area around the cell and work by [48] which could find the cells where there was sufficient contrast between the cell and the background and where they were thick enough.

Having corrected the lighting inhomogeneity, we then tested differing levels of brightness thresholding to attempt to find one suitable of segmenting the lamellopodia. As discussed in Section 4.5, some areas of the cell are brighter and some are darker than the

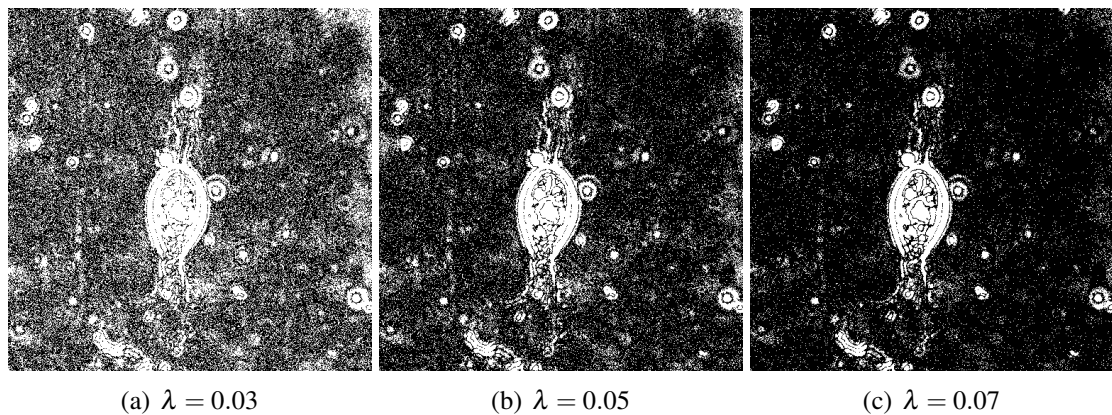(a) $\lambda = 0.03$　　　　(b) $\lambda = 0.05$　　　　(c) $\lambda = 0.07$

Figure 5.17: **Testing Thresholding Levels.** The images all show thresholding of values within a certain percentage of the mean. It can be seen that although the amount background detected increases as we alter the threshold we do not gain any further segmentation of the cell membranes.

background, so to find the cell body we look for those cells that are not within a percentage of the background level (calculated from the image after application of a Gaussian blur). We take the background image as the mean of **G** and denote this $\mu_{\mathbf{G}}$ and take our percentage to be λ, so that the thresholded image $\mathbf{T} = [(\mathbf{I} > \mu_{\mathbf{G}} * (1 + \lambda)) \vee (\mathbf{I} < \mu_{\mathbf{G}} * (1 - \lambda))]$.

In Section 4.5 we found that a λ of 0.1 was most suitable to detect the cell body, but wished to test if levels of thresholding closer to the background level were able to include more of the information. This was tested from 0.01 to 0.1, and three of the results are shown above, in Figure 5.17. It can be seen from Figure 5.17(a) that even at 0.03 we have included a lot of the background in the segmentation but still have not achieved further segmentation of the cell membranes.

5.10.2 Active Contours

The active contour model is a method for segmentation using a curve which deforms to fit the boundary of a shape. The 'snake' of the curve deforms to find the best fit based on parameters which constrain how smooth or jagged the shape should be [1, 11, 55]. We tested various Matlab implementation of this method, finding the most successful to be

(a) Large Square

(b) Small Square

(c) Automated Cell Body

(d) Manual Outline Outside Cell

(e) Manual Outline Inside Cell

(f) Manual Rough Segmentation
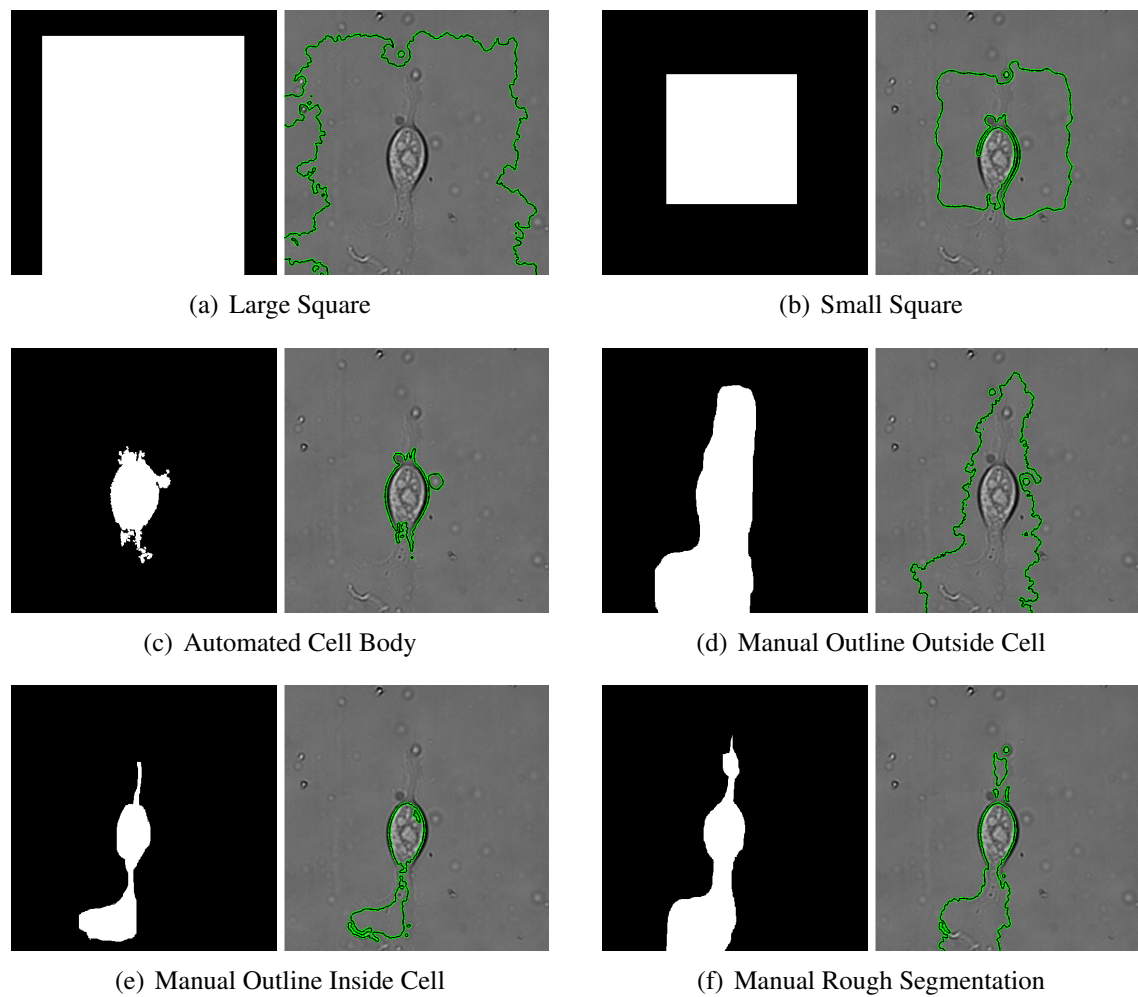
Figure 5.18: **Testing Templates For Active Contours.** The left column shows the starting template for the 'snake' for the active contours method. The second column shows the result as a black line on a green background.

from [27].

We firstly tested using a large square (Figure 5.18(a)) as the initial configuration for the active contour, which surrounded the entire cell and with a smaller square (Figure 5.18(b)) roughly centred on the cell which crosses its boundaries. The larger square only adjusted slightly towards some variations in the background intensity whereas the smaller square gravitated towards the higher contrast boundary of the cell body but do not find the edges or the lamellipodia.

In Figure 5.18(c) we can see the result having provided the system with the cell body

(a) Manually Created Template   (b) 1 (smoothest curve)   (c) 0.5

(d) 0.1   (e) 0.05   (f) 0.001 (roughest curve)
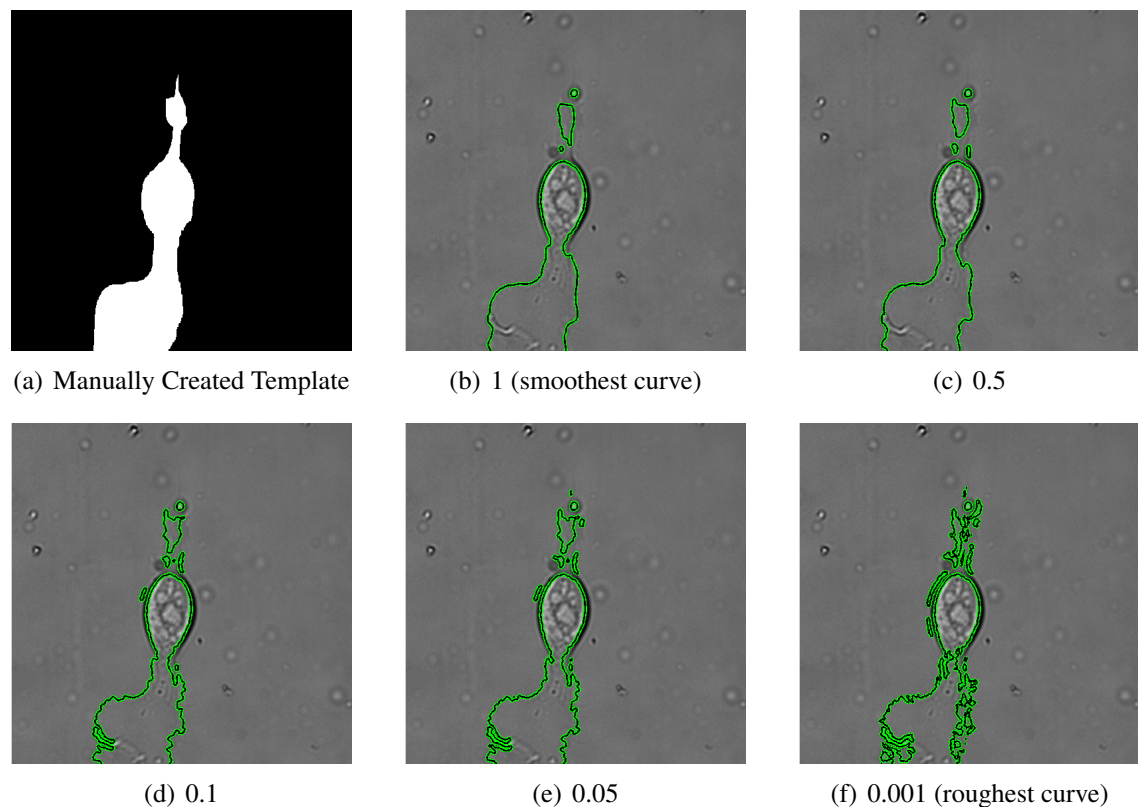
Figure 5.19: **Testing Levels of Smoothing for Active Contours.** We tested various levels of 'smoothness' when using active contours, from 1, which is the smoothest, to 0.001.

segmented by our thresholding (**T**). This was to test what information the method cold provide given an automated template. Although this could converge on the cell body smoothly, and could find some of the different contrast parts of the body with higher iterations, it still did not segment the lamellipodia.

To test what information can be gained from a manually created template we created three outlines based on our cell shape. The first of these is slightly outside of the known cell boundary (Figure 5.18(d)), the second is slightly inside (Figure 5.18(d)) and the final template is a rough manual segmentation of the cell. The template outside of the cell boundary spread further outwards rather than tightening towards the edges, and the template inside the boundary moved further inwards, although it did detect higher contrast substructures in the cell body.

The closest to the cell boundary is the template which starts as a rough manual segmentation of the cell, and is therefore on the boundary to start with. We wish to know if this is providing us with any further information than the manual segmentation alone. All of the results in Figure 5.18 are run with a smoothness of 0.2, which is the default. We tested this with variations on this parameter to try to find more accurate detail than we are providing to the system, with our manually drawn boundary as our initial contour.

Figure 5.19(b) shows the result of using the highest level of smoothing. In this case the lower boundary has moved slightly towards the higher contrast ruffling of the lamellopodia at the bottom and has separated the lamellipodia from the cell body at the top of the cell. Decreasing the smoothness parameter does not provide any better segmentation of the lamellipodia, but gives a less clear outline of the cell body.

From these results (Figures 5.18 and 5.19) we can see that we cannot get a good result from using active contours using an automated template, even if this surrounds the entire cell or already segments part of the cell, such as the body. We have also shown that even with a manually created template that roughly segments the cell body using active contours does not provide us with any additional information.

5.10.3   Texture

To analyse the texture we used a grey-level co-occurence matrix (GLCM), which is a measure of how frequently certain pixels values (in bins) occur within a distance from other values. We created patches over our image and created GLCMs over these. We then obtained the properties of these matrices; contrast, correlation, energy and homogeneity. We then clustered these values using K-means clustering to group together similar patches of texture. To display these images visually a colour was assigned to each patch.

This method was tested using different sized patches and different numbers of clusters, but was not able to detect the lamellopodia. Two examples are shown in Figure 5.20. The first (Figure 5.20(b)) uses a very small patch and only 2 clusters. Although we would like

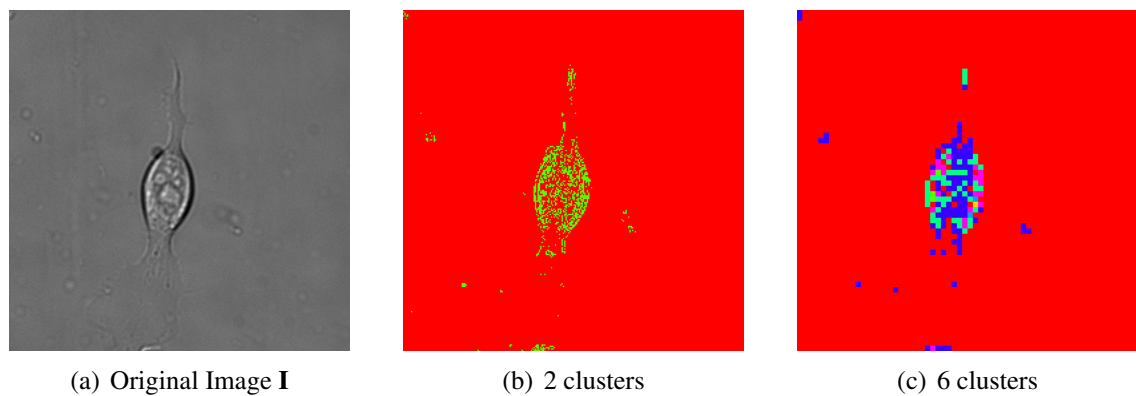(a) Original Image **I**   (b) 2 clusters   (c) 6 clusters

Figure 5.20: **Examples of Texture Analysis to Segment Lamellopodia.** We show two sample results from analysing the texture, one of which uses 6 texture clusters and one uses 2. Both have GLCMs created over different sized patches.

this to separate the cell from the background it has only segmented the cell body and has classified the lamellopodia as being more similar to the background than to the rest of the cell.

In the second example, Figure 5.20(c), we have used 6 clusters which are slightly larger than in the previous example. This again has not been able to classify the lamellipodia, other than a small number of high contrast areas such as the ruffling at the bottom of the image.

This method was tested using patches from 2×2 to 32×32 pixels and using between two and eight clusters, but was not able to detect the lamellipodia or provide additional information than our thresholding stage (Section 4.5).

As we could not use texture clustering to segment the cell we then tried to use this method on cells already segmented by our method, with the background set to black, so that $\mathbf{I} = 0$ where $\mathbf{C} = 0$, as seen in Figure 5.21 (a). We wished to see if the different textures within the cell could enable us to differentiate between the lamellipodia and cell body.

We show two example of this in Figure 5.21. The first image, (b), shows large 16×16 pixel patches and four clusters. Although part of the lamellipodia is showing the same

| (a) Pre-segmented Image | (b) 16×16 Pixel Patches | (c) 2×2 Pixel Patches |
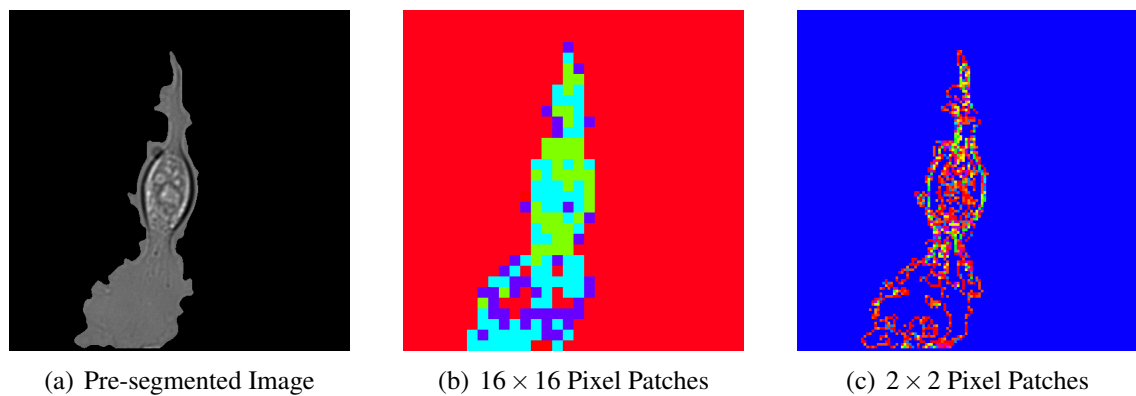
Figure 5.21: **Examples of Texture Analysis to Separate Cell Parts.** We tested the texture clustering method on already segmented cells to try to differentiate between cell parts such as the body and the membranes.

texture as the background (red) and there is slightly more of the yellow cluster in the body, we cannot use this information to assist with further segmenting the cell parts.

The second example shown, Figure 5.21(c), uses the smallest 2×2 pixel patches and eight different texture clusters. The textures can be seen to vary more in the cell body and there is more of the darker blue background colour shown in the lamellopodia, but again we cannot use this to provide us with additional information.

Chapter 6

Segmenting Filopodia

In this chapter we will discuss the appearance and movement of filopodia, and the reasons why we wish to segment these from the cell. We will show some of the problems in achieving this segmentation due to the fineness of these structures and the nature of DIC microscopy.

We will again build on the results from Chapter 4 which provided us with frames free from lighting intensity variations ($\mathbf{I}$) and with a segmentation of the cell body ($\mathbf{T}$). We will use this to locate the filopodia, which are small hair-like structures that protrude from the main cell body.

6.1   Filopodia

Crawling cells produce filopodia in addition to lamellipodia, as shown in Figure 6.2 [33]. In this case it shows the filopodia extending from the lamellipodia at the leading edge (for the direction of movement) of the cell. These are long hair-like projections which should
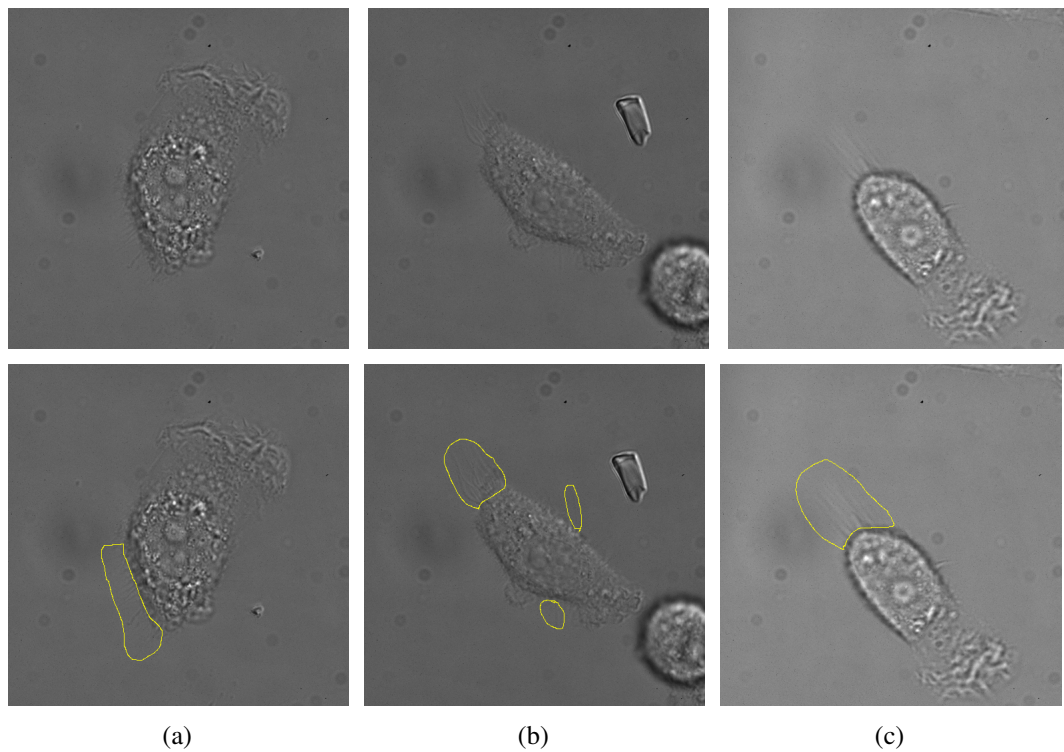
Figure 6.1: **Cells with Filopodia from Three Different Videos.** The top row shows the frames of the video including the filopodia. In the bottom row areas containing the filopodia have been highlighted in yellow.

extend from the leading edge of the cell [8]. They act as feelers, sensing the environment and other cells and guide the cell in which direction to travel [13]. They are a similar thickness to the lamellipodia but are long and thin, rather than being flat and spread out.

6.1.1 The Problem

In Figure 6.3(a) we can see a region of filopodia which we enlarge in Figure 6.3(b). The close up region shows that the filopodia appear to form a stripey pattern but it is difficult to define the edges or to count them individually. In this example the lines are roughly parallel, but in other examples, such as Figure 6.1(b) they curve and do not all form straight lines. This image also shows that they do not always grow in groups but can appear individually at the sides of the cell that are not at the leading edge. These are

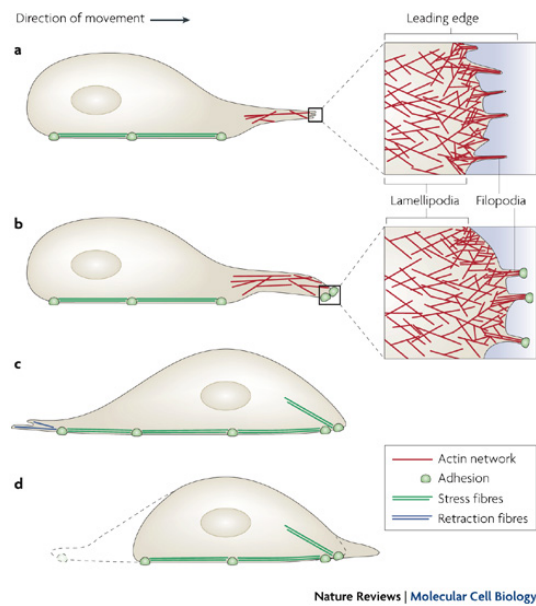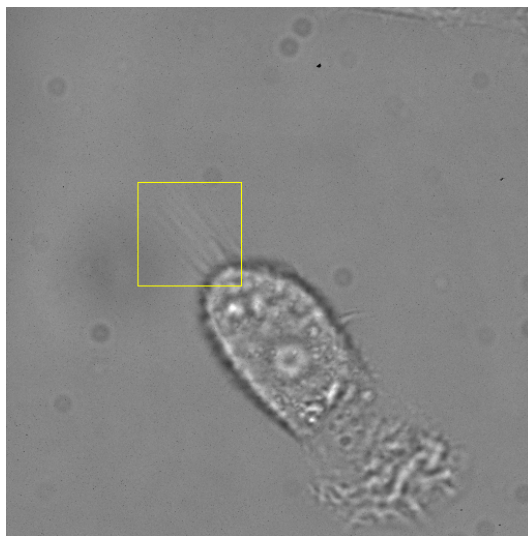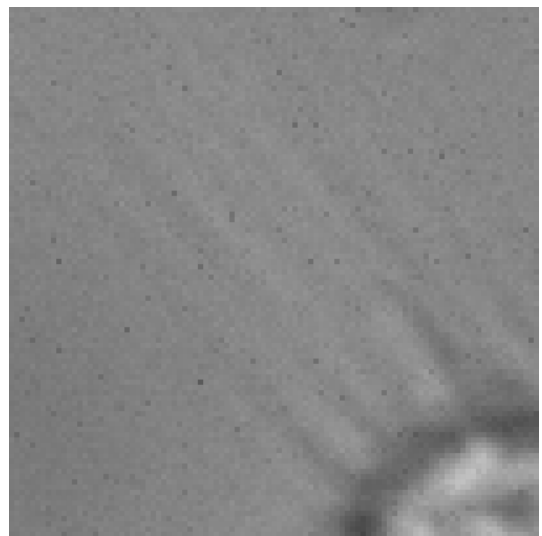Figure 6.2: **A Diagram of Filopodia.** This diagram shows the filopodia extending from the leading edge of the cell [33].

(a) A section including filopodia

(b) A magnification of the area shown in (a)

Figure 6.3: **A Close Up Area of Filopodia** The first image shows a region containing filopodia. This is enlarged in the second image showing the dark and light pattern.

harder to detect by eye and often can only be seen through the movement of the video.

As the work in Chapter 4 has provided us with the location of the cell body (**T**) we can use this information to narrow down the search area as the filopodia is connected to the cell. Initial work on detecting these found that this was not possible using just thresholding and edge detection as they do not show enough difference from the background or have high enough gradients, as shown in Sections 6.4.1 and 6.4.2.

Peak and trough detection was implemented for this region but was not able to detect the filopodia. Hough line detection was also implemented but the 'lines' of the filopodia appeared no more clear than any other shading in the background and the detection was not successful. We also looked individually at the derivative and intensity for manually selected regions with and without filopodia but this was inconclusive. We also looked into summing 'projections', which are the intensities calculated perpendicular to the cell boundary, as discussed in Section 6.4.3, and histograms of pixel values in windows around the filopodia, but were still unable to show the correct filopodia.

Information regarding the intensities over a patch and texture analysis using a grey-level co-occurrence matrix was used in a support vector machine to classify the filopodia but the was also unsuccessful, as discussed in Section 6.4.4.

6.1.2 Biological Significance

Biologists wish to know the effect of knocking down particular genes on filopodia development. The wish to see if the filopodia grow in different places, rather than mainly at the leading edge, and if the length of the individual filopodia changes. This work is done over a series of wild-type videos and a series of knock-down videos. Currently, the filopodia are being manually segmented and counted, which is extremely difficult and time consuming, and subject to error.

6.2 Segmentation Method

We will take a two stage approach to segmenting the filopodia. The first is to locate the areas where the filopodia are likely to be located. The second is then to look for filopodia in these candidate regions and to then segment them individually.

6.2.1 Finding Candidate Regions

(a) $\mathbf{I}(1)$ (b) $\mathbf{F}(1)$ (c) $\mathbf{I}(2)$ (d) $\mathbf{F}(2)$ (e) $\mathbf{I}(3)$
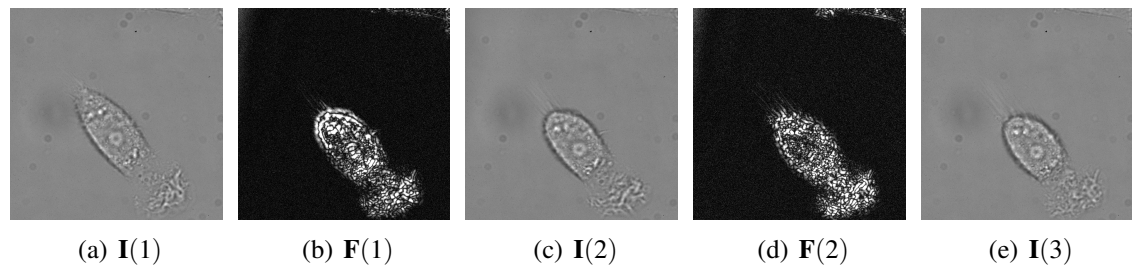
Figure 6.4: **Consecutive Frames and The Differences.** In (a), (c) and (e) we show three consecutive frames from a video and between these we show the differences (contrast adjusted), highlighting the movement between frames. It can be seen that the regions of filopodia are clearer than in the original images.

As discussed in Section 6.1.1, we struggle to see the filopodia in the still images, so, as with the lamellopodia detection (Chapter 5), we will use the information available from the video. We firstly calculate the differences along the third dimension of our video, providing us with a new array showing the movement between the frames, so that $\mathbf{F}(t) = |\mathbf{I}(t) - \mathbf{I}(t+1)|$.

We have shown some examples of this in Figure 6.4, where images (a), (c) and (e) show three consecutive frames from one video. We have taken the differences between and adjusted the resulting values to cover the full range of a grey scale image (from 0 to 1) to show more clearly the movement between the frames. In Figures 6.4(b) and (c) we can see that the areas of filopodia are extending from the top left of the cell. It is more clearly highlighted in these images than in the original frames.

(a) Original Image   (b) Frame Difference   (c) Median Filter   (d) Binary

(e) Remove Isolated Pixels   (f) Increase by 3 Pixels   (g) Decrease by 3 Pixels   (h) Connected To Cell

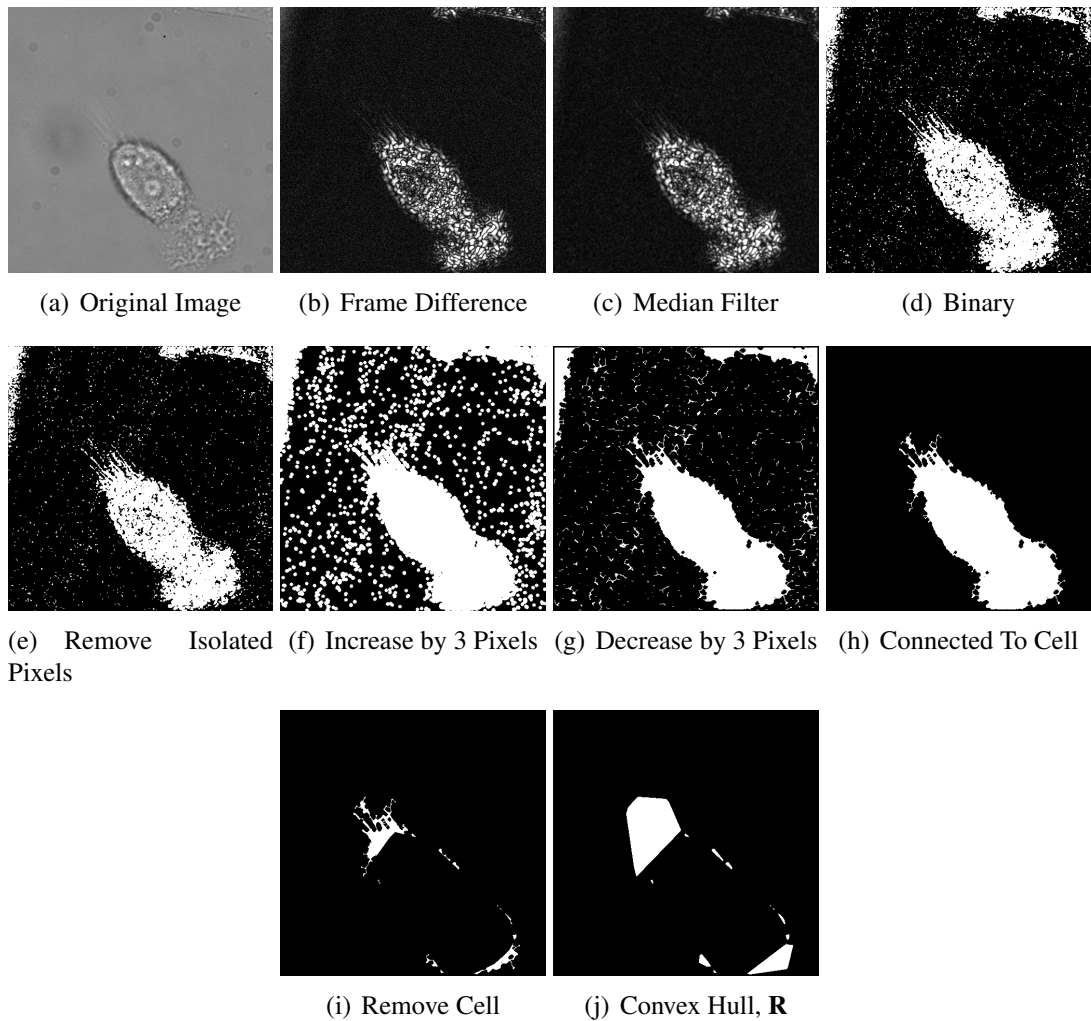(i) Remove Cell   (j) Convex Hull, **R**

Figure 6.5: **Creating Candidate Regions.** We show the stages of creating our initial candidate regions from the difference between frames. The result is a binary image (**R**) of the convex hull of each region.

It can be seen from Figure 6.4 that there are slight changes shown across the majority of the background. From the difference array, we can use thresholding to remove the areas which pick up a tiny amount of change. Examples of this process are shown in Figure 6.5. In Figure 6.5(c) we apply a 3×3 median filter to remove some of the noise from the image. We then threshold this based on the amount of change in the pixel value. Due to the nature of light microscopy and the time lapse imaging most pixels will change a small amount between consecutive frames.

Figure 6.5(d) shows the result of thresholding the image with a level of 0.2, so that the pixels that have changed by equal to or below this level are set to black and the pixels higher to white, resulting in a binary image of the greatest changes.

We next remove the isolated pixels, which are the individual ones surrounded by zeros, as can be seen in Figure 6.5(e). This removes more of the noise from the image. We then use a distance transform to increase each pixel by 3, which connects the nearby pixels, then thin this again by the same amount by removing the external pixels, as can by seen in Figure 6.5(f) and (g).

As we are aware that the filopodia should extend from the cell we remove all regions that are not connected to the cell, as can be seen in Figure 6.5(h). We then wish to 'remove' the cell body from our binary image so that the remaining areas are only those which may contain filopodia Figure 6.5 (i).

Finally, we can create a convex hull around each region of connected pixels individually, so that we have a binary image of candidate regions, as seen in Figure 6.5(j). We will denote our binary candidate region image as **R**.

As we are finding the movement between frames, for those frame differences that are not at the start and the end of the video we can use the information from the prior and the next frame, by combining the candidate regions with those in the current image.
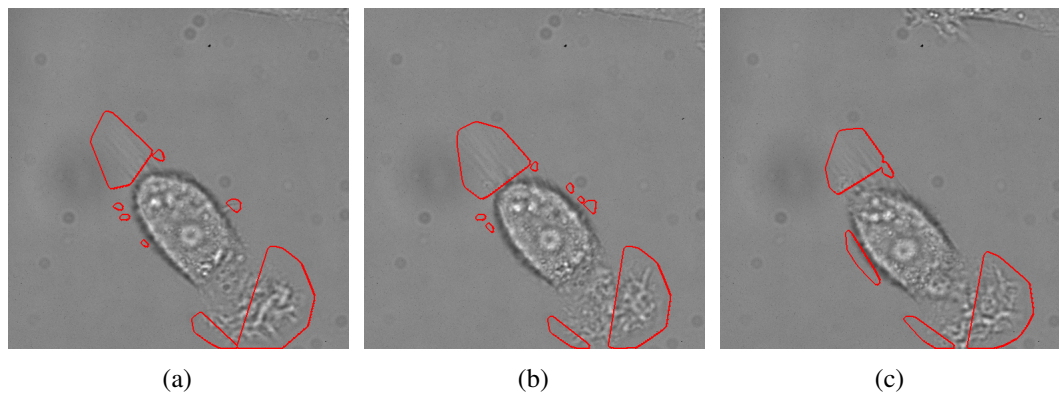
(a) (b) (c)

Figure 6.6: **Initial Candidate Regions.** The images show the candidate regions for filopodia with a red boundary. It can be seen that although the filopodia has been correctly identified, so have other regions of the cell. This is due to cell edges being detected.

Initial Candidate Regions

Some examples of the initial candidate regions are shown in Figure 6.6. We have used the boundary of the convex hulls from the binary image we created to outline these on the original frames. We can see that all areas of filopodia have been identified, including the individual one on the side of Figure 6.6(a). Unfortunately, we have also found a number of regions that are incorrect. As the cell is moving there will be areas around the edge which also show change between frames. When these are connected the convex hull will not just include the edges but also parts of the cell. This can be seen at the lower part of all cells in Figure 6.5, where the edges of the lamellopodia have moved.

Improved Candidate Regions

As some of the initial regions are around the cell border and when surrounded also cover parts of the cell we have already detected, we calculate those regions of which greater than 40% overlaps the cell and remove these from our candidate region set. Some example results are shown in Figure 6.7. In each case we have removed at least one of the incorrect regions, without losing any of the correctly identified regions.

At this stage, we are hoping to have located all the areas potentially containing filopo-
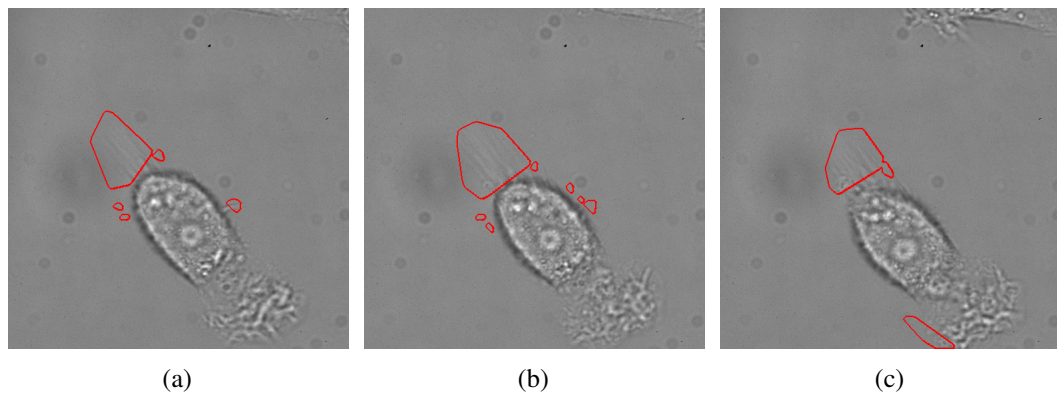
(a) (b) (c)

Figure 6.7: **Improved Candidate Regions.** The images show the new candidate regions for filopodia, again with a red boundary. We have removed some of the incorrect areas which cover larger parts of the cell without losing any of the initial correct regions.

dia so that they can be segmented in the further steps. Manual inspection of each frame from test videos from each set showed that, at this stage, part of each region of filopodia had been detected in 95% of frames. To improve on this, we have again used the information from the surrounding frames, such that if they have a candidate region we should also check for filopodia in that area in the current frame. This increased the regions located to 100%.

6.2.2 Finding Filopodia

The previous Section has provided us with candidate regions in which to check for the existence of filopodia. Figure 6.8(a) shows the detail that is available in the candidate regions when we use our binary image $\mathbf{R}$ as a mask to the rest of the image the mean, so that $\mathbf{I} = \mu_{\mathbf{I}}$ where $\mathbf{R} = 0$. This assists us in the subsequent steps as the intensity variations previously visible in the background would have affected thresholding and other operations on the image. In Figure 6.8(b) the contrast has been enhanced to cover the full range from 0 to 1 to enable easier viewing of the filopodia and other patterns in these regions.

Although the filopodia are small hairlike 'lines', they are rarely completely straight lines. They grow out from the cell but can do so at any angle and are often almost parallel,
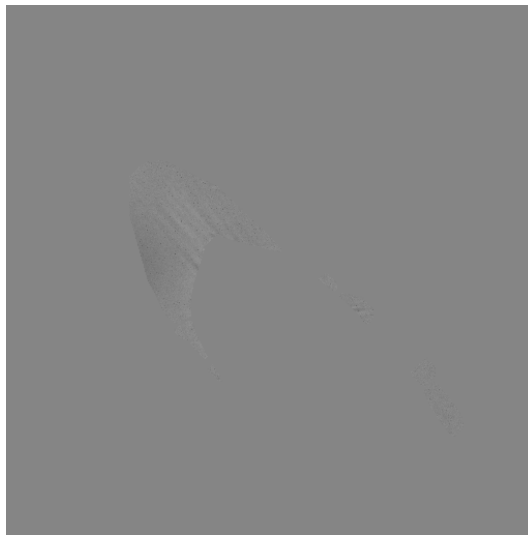
(a) The information from the orginal image in only the candidate regions
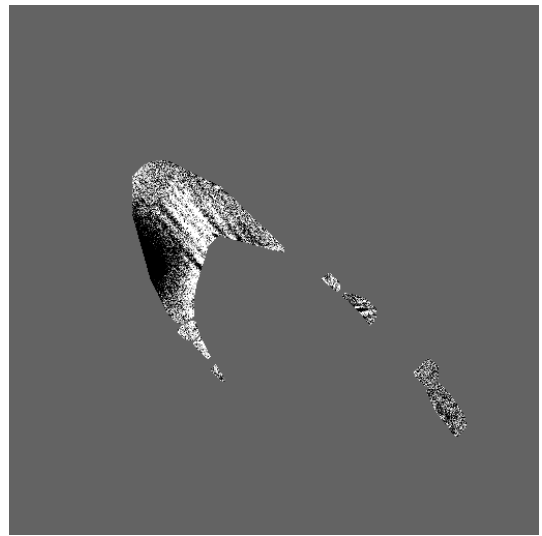
(b) The information from (a) but with the contrast enhanced

Figure 6.8: **Regions of Interest.** The images show all areas except for the candidate regions as set to the mean pixel value. The second image is contrast enhanced to enable easier viewing of the filopodia.

but not enough so that this can be found as a pattern. To find them in the candidate regions we apply morphological opening to a thresholded version of the candidate regions using a short (10 pixel) line shaped structuring element at angles in $10°$ increments.

We firstly threshold our image around the mean, so that our image is set to 1 where $(\mathbf{I} > \mu_{\mathbf{I}} * (1 + \lambda))$ or $(\mathbf{I} < \mu_{\mathbf{I}} * (1 - \lambda))$, using a λ of 0.1. We create templates of zeros the same size as our image and for each angle from $10°$ to $180°$ individually we perform this opening operation to show all of the lines of this length (10 pixels) that could fit entirely within the thresholded image. We then combine the results for every angle to create one binary map of all locations at all angles. The resulting binary pattern shows all of the smaller lines in the image, as can be seen in Figure 6.9 where this has been applied to the candidate regions from Figure 6.8. In the larger regions the lines can be seen crossing each other in multiple directions.

We can now update our candidate regions to remove those where no lines are present. As we are looking for lines growing away from the cell, rather than those lines along the
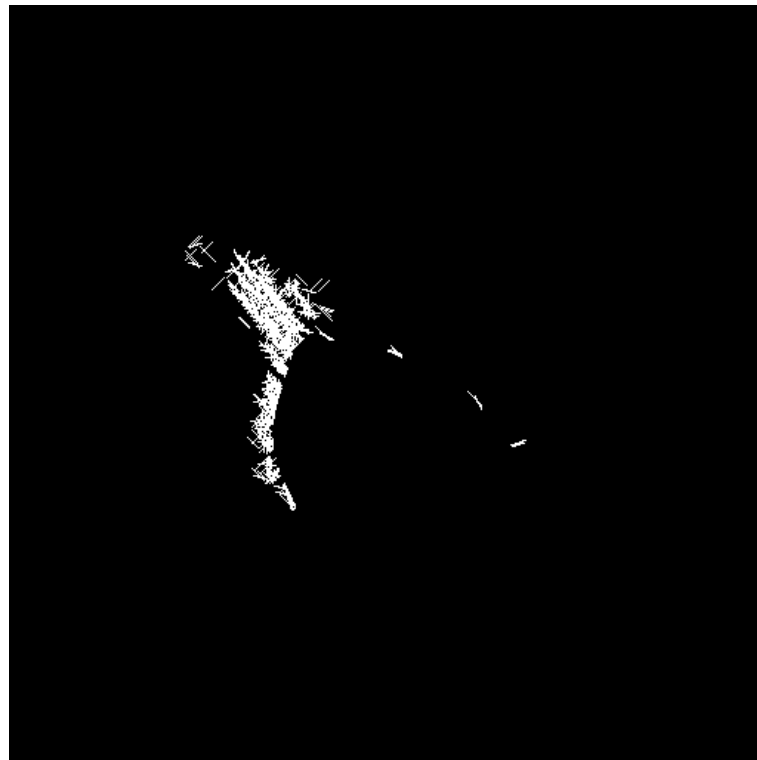
(a) The lines found in the candidate regions

Figure 6.9: **Lines in the Regions of Interest.** We can see all of the shorter lines visible in the regions of interest from Figure 6.8.

cell edge, we take the lines in each of the regions individually and calculate the orientation of these, by using a counter during the opening procedure. We determine the most frequently occurring direction of the lines in each candidate region and then remove those lines which are not roughly at this angle.

We can see the resulting lines in Figure 6.10(a), where the lines in each of the candidate regions are at roughly the same angle. Figure 6.10(b) then shows the result after the removal of those lines which are not connected to the cell body.

From the binary image of the detection of lines growing away from the cell, we can apply an edge detection to find the boundaries of the filopodia. We use the Laplacian of Gaussians method with a threshold of zero so that the result involves closed contours, as it involves all zero crossings from the input image. The outcome is shown in Figure 6.11(a)

(a) All Lines in the Direction of Orientation

(b) Only Lines Which Are Connected To The Body
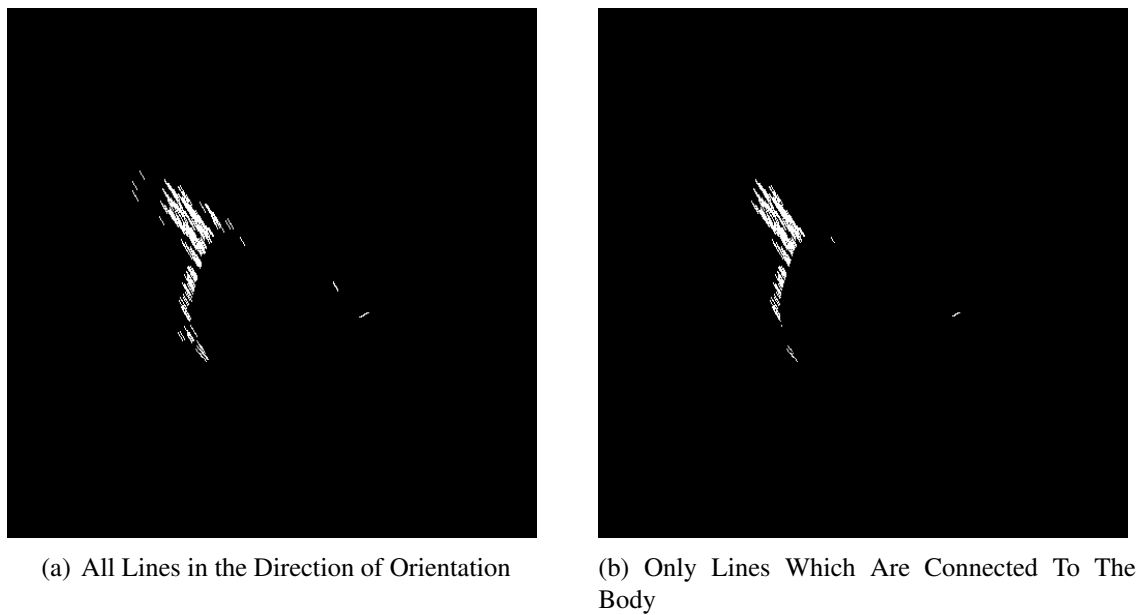
Figure 6.10: **Lines Growing Away From The Cell.** The fist picture shows the lines in the orientation of most of the lines of that region. In the second, those which are not connected to the cell have been removed.

(a) Edges of the Detected Lines

(b) Filled edges
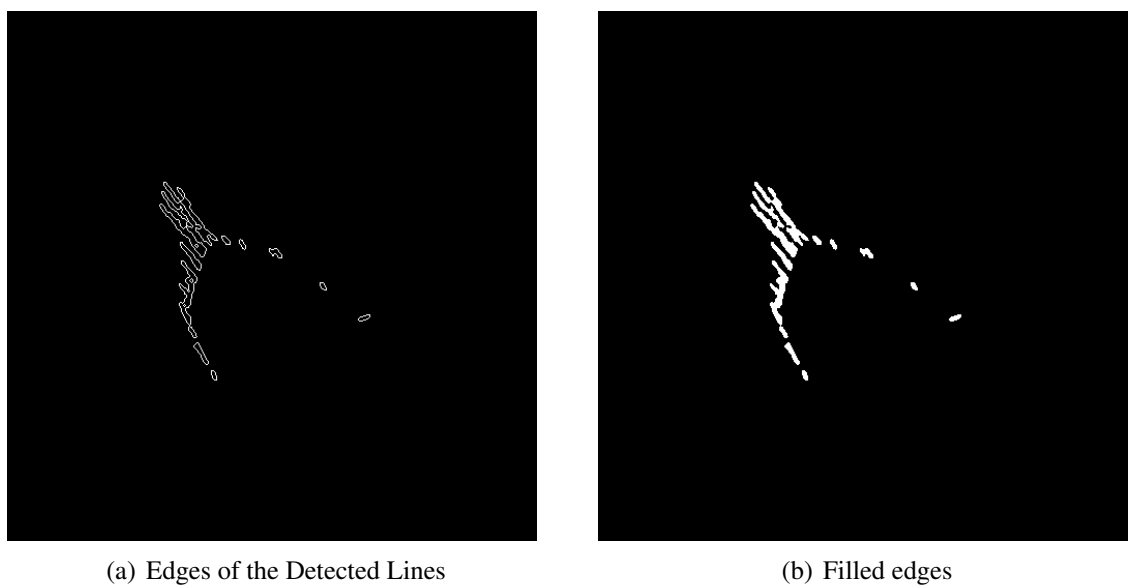
Figure 6.11: **Edges of the Filopodia.** We apply and edge detection to find the edges of the detected lines. We then fill these to show the location of the filopodia.

where we can see the edges of the detected lines. These have then been filled as shown in

Figure 6.11(b).
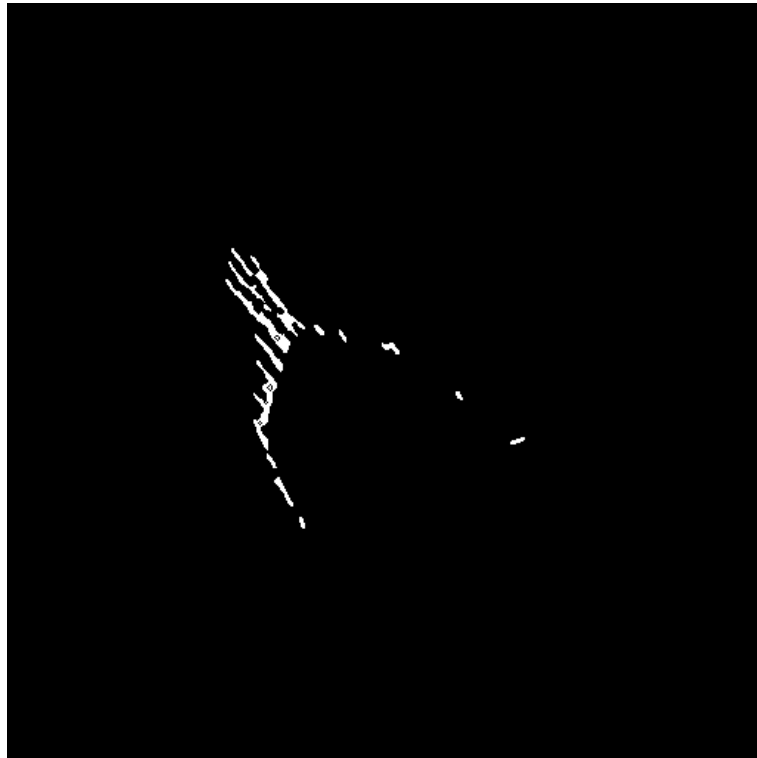
(a) Edges of the Detected Lines

Figure 6.12: **Individual Filopodia.** We remove the edge detection shown in Figure 6.11(a) from the filled image shown in Figure 6.11(b) to separate the individual filopodia to allow counting of the singular structures.

To enable us to view the individual filopodia, rather than the connected area, we remove the edges shown in Figure 6.11(a) from the image shown in Figure 6.11(b) to obtain separated components, as can be seen in Figure 6.12. This now provides us with information regarding the individual filopodia, as we can see their length, location, and whether they are growing in clusters (such as at the leading edge of the cell in the top left) or individually, such as on the right hand side.

6.3   Summary

Filopodia are long, fine protrusions which the cell extends to sense its environment and determine direction of travel. We wished to be able to locate and segment these to de-

termine their length and number, as well as information about where they were growing, such as if they were clustered or singular and if they were more prominent at the leading edge of the cell. As with the lamellipodia, they can be extremely difficult to detect by eye in individual frames.

We looked at the movement between the frames to provide us with candidate areas which may contain filopodia. As we were already aware of the location of the cell body from the preprocessing stages we were able to remove this from our candidate areas. We then included the areas from the previous and subsequent frames, to account for those frames were they did not move or the contrast was too poor for the movement to be detected.

From the candidate areas we were able to look for small line segments which may be part of the larger filopodia. We detected those which grew away from the cell, as we did not want to detect cell boundaries. We were then able to find the edges of these sections and separate them to provide us with individual filopodia. Using this method we were able to find all of the regions containing filopodia, and in those regions provide a degree of segmentation in order to detect if they were growing individually or in groups. Further work still needs to be undertaken to remove the false positive detections of filopodia, as discussed in Section 7.3.

6.4   Other Methods

As we have mentioned, we implemented a number of methods to attempt to segment the filopodia with varying degrees of success. Below we will briefly show why edge detection and brightness thresholding cannot be used to identify these parts and will discuss our implementation of an SVM to try to classify the filopodia.

(a) Cell With Filopodia

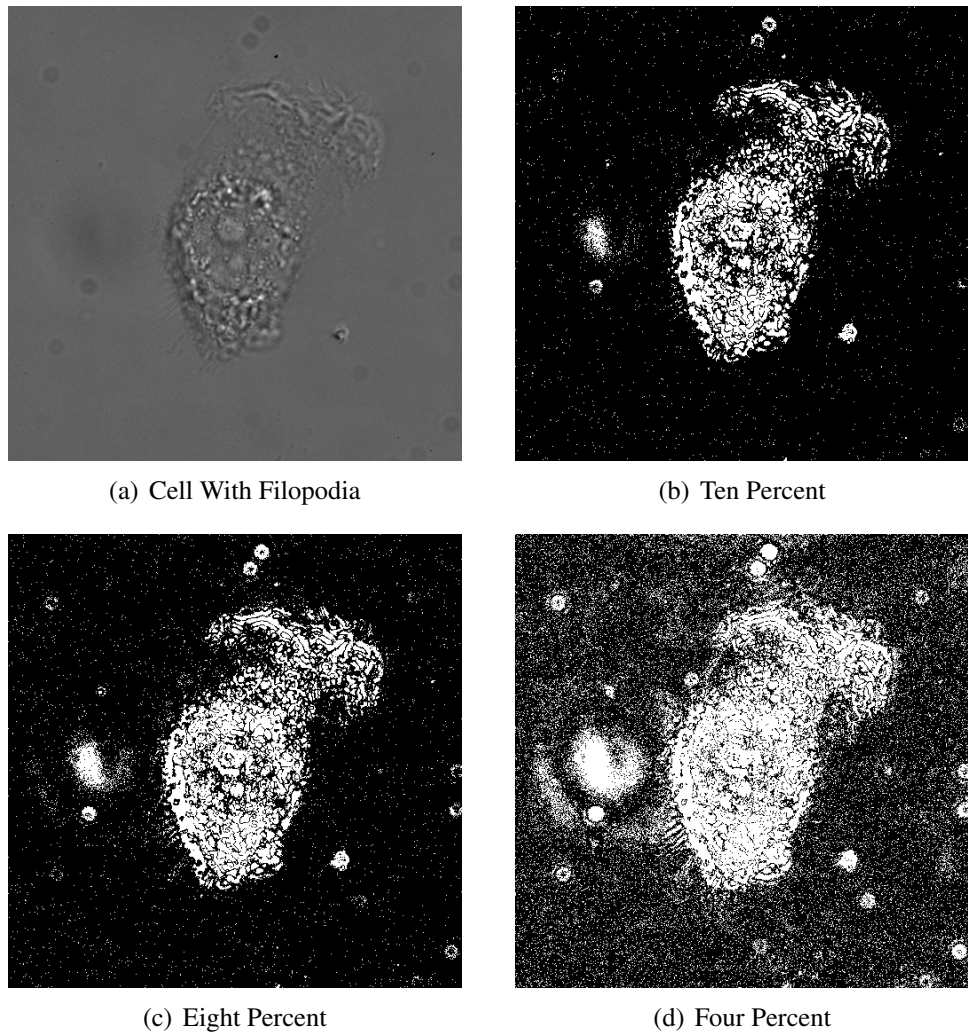(b) Ten Percent

(c) Eight Percent

(d) Four Percent

Figure 6.13: **Thresholding Filopodia Images** We show the results of thresholding image (a) by detecting the pixels brighter and darker than a percentage of the mean image intensity. Using a level that enables us to detect the majority of the filopodia also detects a high proportion of the background so doe not assist with the segmentation.

6.4.1   Brightness Thresholding

We are already using thresholding to provide us with a template for where the cell body is, which is the result $\mathbf{T}$ from our preprocessing steps in Chapter 4. We wished to also check if thresholding could be used to segment the filopodia which are attached to the cell body.

Figure 6.13(a) shows an image with many small filopodia growing from the bottom left of the cell. These are also circled in Figure 6.1(a). We have then shown the result of using three different brightness thresholds. These were applied in the same way as in Section 4.5, so that for each percentage λ we create a thresholding:

$$\mathbf{T} = [(\mathbf{I} > \mu_{\mathbf{I}} * (1 + \lambda)) \vee (\mathbf{I} < \mu_{\mathbf{I}} * (1 - \lambda))].\tag{6.1}$$

The first result shown in Figure 6.13(b) is using a λ of 0.1, as was done to locate and segment the cell body. It can be seen that although there are a few pixels found in the region of the filopodia these have not been identified. The result in (c) shows that with a λ of 8 we have increased the amount of pixels thresholded but there are still only small parts of the filopodia being detected. The final result shows that in order to threshold the majority of the filopodia we also detect a large amount of the background of the image, so this does not assist us with the segmentation.

As the filopodia are often thicker where they are connected to the cell, we can more easily threshold these parts than the hairlike ends, where the contrast is very poor. We also implemented a method which initially found the brightest and darkest pixels attached to the cell, then amended the threshold on each run to look for those pixels which were not as high contrast but were connected to those which had already been identified. Due to these being so close to the levels of intensity of the rest of the image this was also unsuccessful as the detection spread to covering large parts of the background and away from areas of the cell where no filopodia where present.

6.4.2 Edge Detection

When finding the lamellopodia in Chapter 5 we were able to use edge detection to locate these thin membranes which were attached to the cell, so this method was also tested on our filopodia images. We have shown the results of just the edge detection in Figure 6.14, using a Canny edge detection with the upper threshold set to the figure shown, as described in Section 5.3.

It can be seen from the results in Figure 6.14(c), (f) and (i) that if the edge detection threshold is high enough not to detect the background then only very short parts of some of the brighter filopodia are detected. The level at which we could even find these parts varied from image to image and we could not find a method to automate the setting of the edge detection threshold.

6.4.3 Connecting Projections

As the filopodia generally grow outwards away from the cell we implemented a method to sum the calculate values of these projections, hoping that this would provide us with information regarding the darker and brighter lines of the filopodia.

We wished to calculate these values at boundary points increasing in size away from the cell. The aim of the method was to find the points which were dark or bright, then to connect these to find to lines of the filopodia. Figure 6.15 shows the results of this on a test image. The first picture shows in yellow a boundary of an imaginary cell and the other three are the points at which we collect the data. The result in Figure 6.15(b) shows the result of this very simple test showing how that 'significant' pixels are detected. The next stage would be to connect these points to segment the lines.

Figure 6.16 shows the result of testing this on a real cell image. We can see where the data was collected in (a) and (b) shows the points which were detected as being significantly bright or dark. In this particular case a large part of the filopodia were detected, but a higher number of points were collected at other parts of the cell where there were

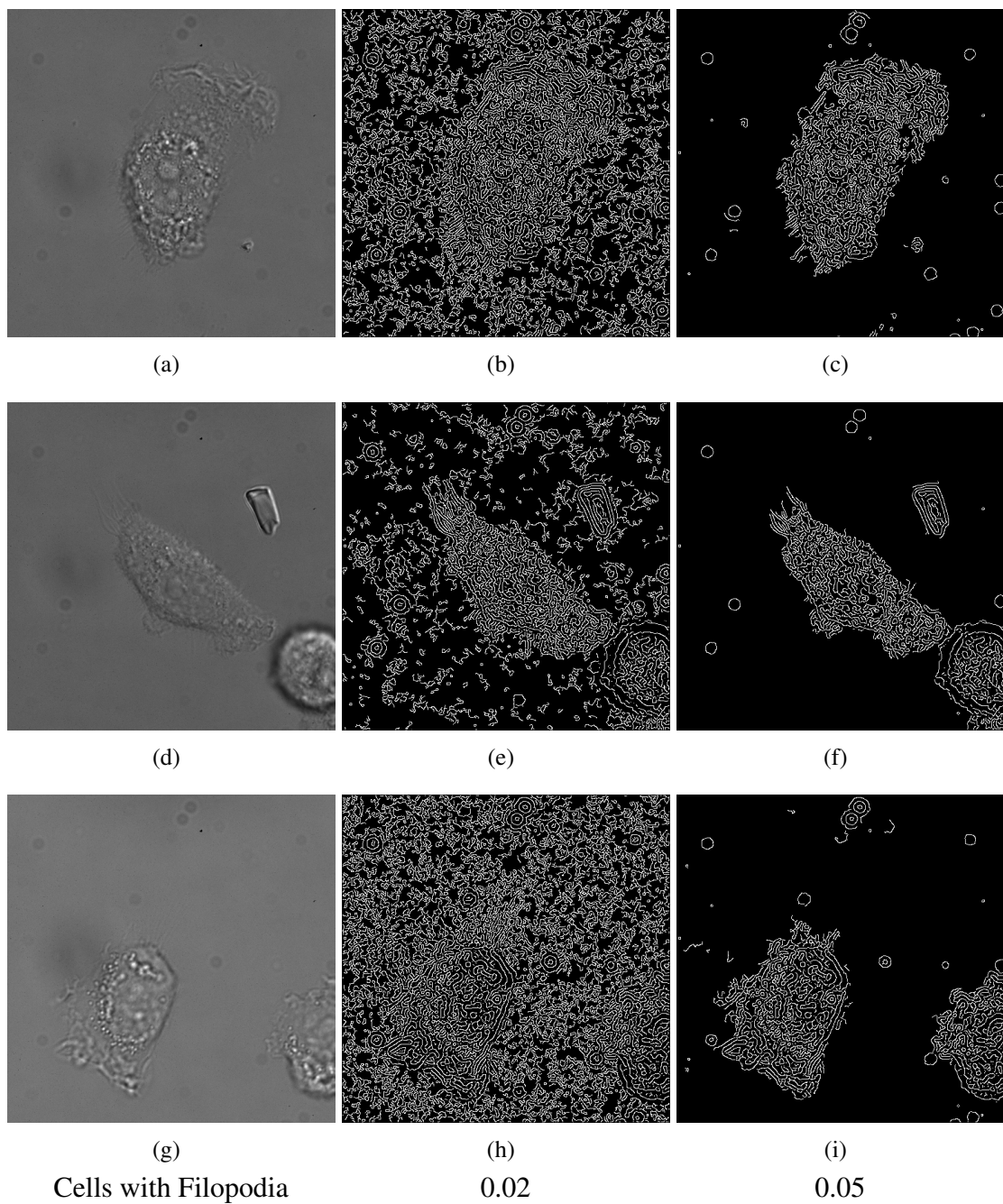|		
(a)	(b)	(c)
(d)	(e)	(f)
(g)	(h)	(i)
Cells with Filopodia	0.02	0.05

Figure 6.14: **Edge Detection on Filopodia Images.** We show the result of running Canny edge detection on a cell with filopodia. With an upper threshold of 0.02 the background is detected and cannot be differentiated from the cell. With a higher threshold we do not detect enough of the filopodia to allow them to be segmented.

(a) Test Borders (b) Test Results
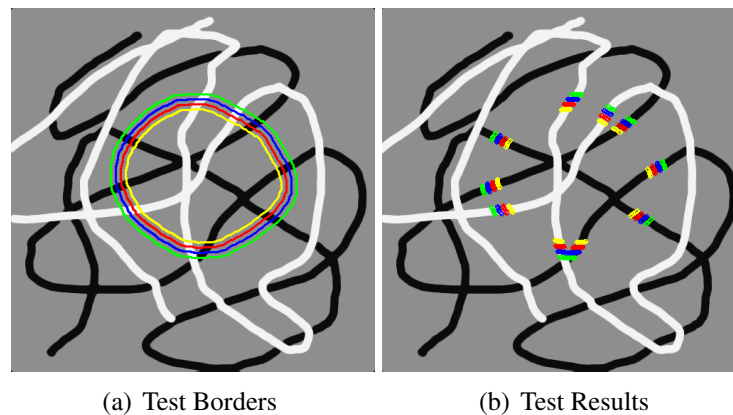
Figure 6.15: **Testing Pixel Summing.** As features in DIC images are both brighter and darker than the background we tested finding these features so that they could be connected to make lines. Our test shows that the brighter and darker parts that fell onto the 'boundaries' could be detected

no filopodia present.

In Figure 6.16(c) we show only the dark pixels which we wished to connect to form lines and in (d) we show only the brighter pixels. From testing this over a number of images we were not able to reliably gain any information about the filopodia location, and the image shown was one of our more successful results.

6.4.4 Support Vector Machine

As we were unable to find a particular characteristic of the filopodia that would allow us to locate them we implemented a support vector machine (SVM) in which we would provide many different types of information in the hope that it would classify our filopodia on test images then allow us to detect these on others.

This was done using a sliding window of varying sizes so that we could obtain information relating to the regions not available from one pixel alone. We manually marked images and set the pixels containing filopodia to 1 and those not containing filopodia to zero. We tested various degrees of filopodia in each sliding window, to classify if it was contained or not, from counting a window as 'on' if it contained only one pixel of filopo-

(a) Cell Boundary and Increasing
Test Points

(b) Results on Real Cell

(c) Dark Pixels
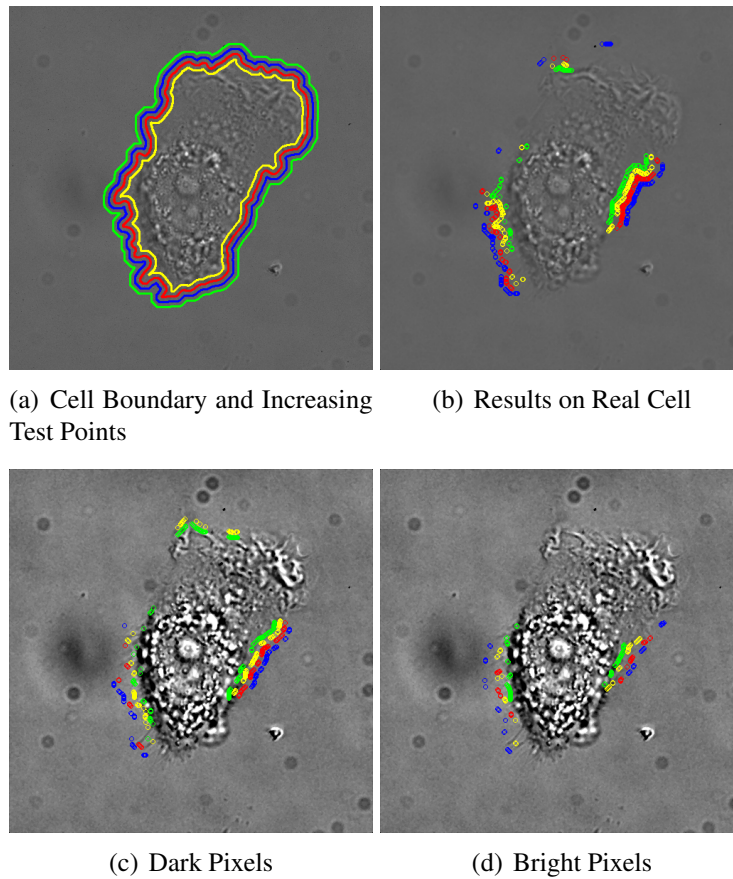
(d) Bright Pixels

Figure 6.16: **Significant Pixels on a Real Image.** Image (a) shows the locations at which
we check for bright and dark pixels, and image (b) shows where these were detected. The
remaining images show the brighter and darker pixels separately.

dia, or the centre pixel, up to counting it as 'off' if more than half of the window was not filopodia.

Information passed to the SVM included the maximum, minimum and mean pixel values in each window, the histogram of the pixel values, the entropy and statistics from a GLCM including the contrast, correlation energy and homogeniety. We tried training and testing on entire images, on small patches of filopodia and on a region of interest around the cell boundary.

None of our tests produced any results that could assist in any way with locating the filopodia. Although some of the test images for which we could provide a manual classification showed the percentage output of correct classified pixel as very high this was only because it had marked so few pixels as filopodia and they take up so little of the image (for example, a result which stated there were no filopodia at all would be around 98% accurate). Despite many tests and variations on our SVM we were not able to produce a result that was better than random.

Chapter 7

Conclusion

In this Chapter we will firstly discuss a summary of what we have covered during this thesis, including our image processing stage and our techniques for segmenting the lamellipodia and filopodia. We will then discuss some of the issues and other work that was undertaken during the period of this research. We will then cover some of the further work which has not yet been undertaken or completed.

7.1 Summary

For hundreds of years, cells have been studied under the microscope. This work is still ongoing and as progressions have been made in technology, so has our ability to see and study the cell in more depth and detail. The invention of differential interference contrast (DIC) microscopy has allowed us to film cells over time, with a non-invasive technique that does not damage the cell or affect its movement and behaviour. As the footage we obtain increases, so does the work taken to abstract all of the necessary data from it. Much

of this work is done by hand, which is time consuming and subject to human error. We wish to automate this work in order to provide biologists with the information they require in order to study normal and abnormal cells, and the changes in behaviour when genes or chemicals are affected.

There are many types of moving cells, including those that swim, those that glide, and those that can crawl over surfaces. The cells we have studied are those that use protrusions to sense the environment and to crawl. These protrusions give the cell an irregular shape that can change dramatically. They are also extremely thin, making them extremely difficult to see by eye. Due to this, we often have to use the information gained over time to assist us in locating and segmenting them.

7.1.1   Image Preprocessing

As DIC microscopy causes lighting variations across the videos, giving the effect of being lit from one particular area and darker in other areas, there are problems in thresholding the images and applying other image processing techniques. To resolve this we preprocess our images to remove this, using an extremely efficient technique which requires no parameter setting as it is based purely on the size of the input image. This was shown to work effectively not only on our own videos but also on DIC images acquired from an online search. As there was no ground truth for these images, to test our results we created an image with no background lighting variations and manually added a shadow and light effect to it. We ran our method on both the shaded and the unshaded image and by subtraction of the resulting frames from the original image we proved that our method can remove the added shading with no loss of detail, but would also not affect an unshaded image.

The lighting correction technique then enabled us to apply brightness thresholding which was not previously possible. This then provided us with the main cell body as well as information about the sizes of the cells, which was used to remove non-cellular

material from the images, both at this stage and in later stages of the process.

7.1.2   Segmenting Lamellipodia

Lamellipodia are large, thin membranes which the cell extends to assist with crawling. They adhere to substrate at the leading and rear edge of the cell and both push and pull the cell forward. We wished to be able to segment these with a boundary as close as possible to the cell edge, to provide information regarding the size and shape. This was initially done by finding the edges connected to the cell body, without any need for manual selection or parameterisation. When tested against the leading software for segmenting cells with parameters optimised for each video to give the most accurate segmentation, our method proved to be more accurate in the majority of cases.

From looking at the images where our segmentation was poor, it was clear that the edges in the still frame were not visible by eye and could only be detected when watching the moving cell. To improve our method we took the information from the surrounding frames. We calculated the difference in cell size between frames, using those frames where the size had changed dramatically to indicate that a problem had occurred in the segmentation. We then looked at the intersection of the frames around this and reapplied the edges to give the detail of the cell shape. This was done iteratively until no changes were made or the cells did not change dramatically. We again tested against the previously mentioned software and our accuracy had improved so that we were more accurate for every set of videos, at around 90% using the DICE similarity score.

There were still a few frames that had a poor segmentation due to low edge strength, using both our method and the manually parameterised output. We found that we could detect the cases where the entire frame had been detected, and in these cases used purely the information from the surrounding frames, rather than the edges we had detected in the current frame. This method increased the accuracy of the segmentation but was very much a 'best guess' method, as we assume that the cell is somewhere between the pre-

ceding and subsequent frame. Further ideas to improve this are suggested in Section 7.3, although the segmentation is currently extremely accurate for all but a couple of frames. We additionally tested this method on videos with no lamellipodia present and showed that there was no loss of accuracy in their segmentation.

7.1.3   Segmenting Filopodia

Filopodia are long, fine protrusions which the cell extends to sense its environment and determine direction of travel. We wished to be able to locate and segment these to determine their length and number, as well as information about where they were growing, such as if they were clustered or singular and if they were more prominant at the leading edge of the cell. As with the lamellipodia, they can be extremely difficult to detect by eye in individual frames.

We looked at the movement between the frames to provide us with candidate areas which may contain filopodia. As we were already aware of the location of the cell body from the preprocessing stages we were able to remove this from our candidate areas. We then included the areas from the previous and subsequent frame, to account for those frames where they did not move or the contrast was too poor for the movement to be detected.

From the candidate areas we were able to look for small line segments which may be part of the larger filopodia. We detected those which grew away from the cell, as we did not want to detect cell boundaries. We were then able to find the edges of these sections and separate them to provide us with individual filopodia. Using this method we were able to find all of the regions containing filopodia, and in those regions provide a degree of segmentation in order to detect if they were growing individually or in groups. We could also determine the rough length of these for each region. Further work still needs to be undertaken to remove the false positive detections of filopodia, as discussed in Section 7.3.

7.2   Discussion

The initial step of this work was to find and locate all parts of a cell, even when thin membranes were present. Unfortunately, none of the other methods found were able to do this. Trying methods such as edge detection alone could sometimes provide a good boundary of the cell, but the fine tuning necessary to obtain this was often more time consuming than undertaking the work manually. It also differed for every video, and often between frames. This was also the problem with the existing programs that we tried. There are some which allow a series of steps to be built up but again this takes a lot of time and knowledge of the meaning of all of the parameters.

Due to this, we wished to find a method that could do this automatically. Our aim was for full automation, such that the only thing we need to provide to the system is the video or image. We were not able to find any other method which claimed it could do this for irregularly shaped cells or those with thin membranes. The segmentation system described in Chapters 4 and 5 was tested on many cell videos other than the ones described in this thesis to ensure that we could still use it to segment more simply shaped cells or those that could already be detected by thresholding alone, such as cells without lamellipodia.

Many methods were tried prior to the iterative implementation we have used. Active contours proved not to be suitable as the settings required to detect the fine edges of the lamellipodia and the frequent changes of direction meant that the same settings would then not detect the cell body correctly, and conversely, detecting the cell body succesfully resulted in the contour not including the thinner membranes. Although we considered using seperate parameters for the body and for the lamellipodia, we still needed knowledge of where the lamellipodia was occuring and this again required different settings for each video.

We also tried using texture details from a grey level coocurrence matrix (GLCM) and clustering these to find similar textures, but this was again able to work for the cell

body and not for the full boundary. Although some of the more obvious edges could be classified we were not able to create a segmentation from this.

Segmenting the filopodia was tackled as a separate piece of work, rather than as part of the cell detection system. This is because we are not just trying to find a boundary that surrounds these but to be able to seperate them from all other parts of the cell, including the cell body and the regions of 'blebs'. We were unable to find any existing method that was able to use image processing techniques to do this. The only works found that involved filopodia simply detected the region they might be occurring, but this appeared to involve a larger border around the cell and there was no technique to account for the areas where there were no filopodia or to separate them.

We did not wish to just find a possible region for the filopodia, as previous methods have done, as this would simply have involved using a larger border than the cell as we are aware that they grow from the edge of the cell. We used small candidate regions so that we could find where on the cell they were growing from and to be able to detect if they were individual or growing in clusters. Alternative methods tried included those used previously, such as active contours and texture analysis but these could not detect these areas. We were also unable to segment them using edge detection, even when this was parameterised to find the closest match. We also tried using a Support Vector Machine (SVM) to classify these regions using as much information as possible. This was implemented using a moving window of small, overlapping patches. We included information on the pixel intensity, edge strength, GLCM details and histograms of pixel values. This was also unable to classify the filopodia, even when we removed all detail of the cell body from the image.

7.3   Further Work

Although there are still many advancements to be made in this field, the further work we discuss here is mainly based on methods that were not completed during the course of this research.

7.3.1   Further Segmentation of Lamellipodia

As we discussed in Chapter 5.8, we obtained an extremely accurate segmentation for the majority of frames, but there were still a few with problems. The method implemented when the edge detection was not at an appropriate threshold for the frame increased the accuracy but was not able to use any of the information from the current frame. One method would be to try to fix a boundary half way between the previous frame and the next frame. Various methods were tried for this, including the watershed method, level sets, and trying to find points around both boundaries so that the half way position could be calculated. This was not particularly accurate as often the boundaries did not align well so points could not be matched.

A better solution would be to try to alter the edge detection itself. To do this we would again need to take measurements of drastic cell size change between frames. If the cell was reported as being too large we could increase the edge detection threshold to detect fewer edges so that our 'filling' does not spread to cover the background. Additionally, if the cell is reported as being too small we could lower the threshold to detect the finer edges around the lamellipodia.

Unfortunately, this method requires knowledge of which is the correct cell size. If our first frame is incorrect we may be adjusting all subsequent frames to be a similar size detection to the initial bad segmentation. Observing the first frame to ensure that it is correct would involve manual input, which we have so far managed to not include in this work. We are aware of what the cell body size should be, as this is recorded in the

preprocessing steps and is used to remove non-cellular material from the images, but this does not necessarily advise us how large an area should covered with lamellipodia as they can sometimes cover very small areas as they start to protrude and sometimes cover a large proportion of the frame, and grow much larger than the cell body.

7.3.2 Further Segmentation of Filopodia

As discussed in Chapter 6.2.2, we are able to find all regions which include filopodia, and from these determine the length of these, the location around the cell, and whether they are growing individually or in a cluster. Unfortunately, we are still left with some false positive classifications. This can occur from small marks on the substrate or from the microscope, or from shadow from the cell body. To remove those along the cell boundary we could again look at the direction of the region overall. After we have separated the filopodia we calculate if they are growing away from the cell or along it and remove those which do not grow away. To remove the detections due to other objects or marks on the image we would need to look further into the shape of the filopodia that are detected.

Bibliography

[1] Active contours without edges. *Image Processing, IEEE Transactions on*, 10(2):266–277, February 2001.

[2] M. D. Abramoff, P. J. Magelhaes, and S. J. Ram. Image processing with ImageJ. *Biophotonics Int*, 11(7):36–42, 2004.

[3] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell, Fourth Edition*. Garland Science, 4 edition, 2002.

[4] Revathi Ananthakrishnan and Allen Ehrlicher. The forces behind cell movement. *International journal of biological sciences*, 3(5):303–317, 2007.

[5] W.M. Becker, L.J. Kleinsmith, and J. Hardin. *The World of the Cell*. Benjamin/Cummings series in the life sciences. Benjamin Cummings, 2000.

[6] Ryoma Bise, Kang Li, Sungeun Eom, and Takeo Kanade. Reliably tracking partially overlapping neural stem cells in dic microscopy image sequences. In *MICCAI Workshop on Optical Tissue Image Analysis in Microscopy, Histopathology and Endoscopy (OPTMHisE)*, pages 67–77, September 2009.

[7] E. Brannock, M. Weeks, and V. Rehder. Detecting filopodia with wavelets. In *Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on*, pages 4 pp.–4049. IEEE, May 2006.

[8] D. Bray. *Cell Movements: From Molecules to Motility*. Garland Pub., 2001.

[9] John Canny. A Computational Approach to Edge Detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698, November 1986.

[10] Tony Collins. Wright cell imaging facility. `http://www.uhnres.utoronto.ca/facilities/wcif/fdownload.html`. [Online] Accessed 04 December 2104.

[11] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models-Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.

[12] Cronodon. Cell locomotion - the cytoskeleton and cell crawling. `http://cronodon.com/BioTech/Cells_motility_3.html`, 2014. [Online] Accessed 31 December 2014.

[13] J. Davies. *Mechanisms of Morphogenesis*. Elsevier Science, 2005.

[14] Dirk Dormann and Cornelis J. Weijer. Review: Imaging Of Cell Migration. *European Molecular Biology Organization Journal*, 25(15):3480 – 3493, August 2006.

[15] Graham A. Dunn and Gareth E. Jones. Cell motility under the microscope: Vorsprung durch Technik. *Nature Reviews Molecular Cell Biology*, 5(8):667–672, August 2004.

[16] C.A. Glasbey and G.W. Horgan. *Image analysis for the biological sciences*. Statistics in practice. J. Wiley, 1995.

[17] L. Gledhill. Heart of a snowflake using dic microscopy. `https://www.flickr.com/photos/13084997@N03/8417795139/`, January 2003. [Online] Accessed 01 December 2014.

[18] GE Healthcare. Deltavision microscopy imaging systems. `http://www.gelifesciences.com/webapp/wcs/stores/servlet/CategoryDisplay?categoryId=1171164&catalogId=70101&productId=&top=Y&storeId=12751&langId=-1`, 2015. [Online] Accessed 05 July 2015.

[19] Marc Herant and Micah Dembo. Form and function in cell motility: from fibroblasts to keratocytes. *Biophysical journal*, 98(8):1408–1417, April 2010.

[20] R.M. Jiang, D. Crookes, Nie Luo, and M.W. Davidson. Live-Cell Tracking Using SIFT Features in DIC Microscopic Videos. *Biomedical Engineering, IEEE Transactions on*, 57(9):2219–2228, September 2010.

[21] G. Karp. *Cell Biology*. John Wiley and Sons, Asia, sixth edition, 2010.

[22] K. Keren, Z. Pincus, G.M. Allen, E.L. Barnhart, G. Marriott, A. Mogilner, and J.A. Theriot. Mechanism of shape determination in motile cells. *Nature*, 453(7194):475–480, May 2008.

[23] N. Kerrison. Automated analysis of cell motility using computer vision techniques. M.res dissertation, University of Leeds, 2010.

[24] John Kolega. The Role of Myosin II Motor Activity in Distributing Myosin Asymmetrically and Coupling Protrusive Activity to Cell Translocation. *Molecular Biology of the Cell*, 17(10):4435–4445, October 2006.

[25] A. Kuijper and B. Heise. An Automatic Cell Segmentation Method for Differential Interference Contrast Microscopy. In *ICPR 2008. 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, December 2008.

[26] Jon D. Lane and Howard Stebbings. Transmitted light imaging. In D. Stephens, editor, *Cell Imaging*, Methods Express. Scion, Bloxham, Oxfordshire, 2006.

[27] Shawn Lankton. Active contours. `http://www.shawnlankton.com/2007/05/active-contours/`, May 2007. [Online] Accessed 01 July 2015.

[28] Kang Li and Takeo Kanade. Nonnegative mixed-norm preconditioning for microscopy image segmentation. In *Proceedings of the 21st Biennial International Conference on Information Processing in Medical Imaging (IPMI)*, pages 362–373, July 2009.

[29] Kang Li, Eric D. Miller, Lee E. Weiss, Phil G. Campbell, and Takeo Kanade. Online tracking of migrating and proliferating cells imaged with phase-contrast microscopy. In *IN: PROC. IEEE CONF. COMP. VISION AND PATT. RECOG. WORKSHOP, P. 65. IEEE COMPUTER SOCIETY PRESS, LOS ALAMITOS*, 2006.

[30] John Limouze, Aaron F. Straight, Timothy Mitchison, and James R. Sellers. Specificity of blebbistatin, an inhibitor of myosin II. *Journal of muscle research and cell motility*, 25(4-5):337–341, 2004.

[31] Masur. S. cerevisiae under dic microscope. `http://en.wikipedia.org/wiki/Phase_contrast_microscopy`, January 2010. [Online] Accessed 01 December 2014.

[32] MATLAB. *version 8.0.0 (R2012b)*. The MathWorks Inc., Natick, Massachusetts, August 2012.

[33] Pieta K. Mattila and Pekka Lappalainen. Filopodia: molecular architecture and cellular functions. *Nat Rev Mol Cell Biol*, 9(6):446–454, June 2008.

[34] C.A. Middleton and J.A Sharp. *Cell Locomotion in Vitro: Techniques and Observations*. Croom Helm Ltd, Kent, 1984.

[35] T. J. Mitchison and L. P. Cramer. Actin-based cell motility and cell locomotion. *Cell*, 84(3):371–379, February 1996.

[36] Alex Mogilner and Kinneret Keren. The Shape of Motile Cells. *Current Biology*, 19(17):R762–R771, December 2014.

[37] D. P. Mukherjee, N. Ray, and S. T. Acton. Level set analysis for leukocyte detection and tracking. *Image Processing, IEEE Transactions on*, 13(4):562–572, April 2004.

[38] Douglas B. Murphy. *Fundamentals of Light Microscopy and Electronic Imaging*, chapter Differential Interference Contrast (DIC) Microscopy and Modulation Contrast Microscopy, pages 153 – 172. Wiley-Liss, New York, New York, 2001.

[39] Z. Pincus and J. A. Theriot. Comparison of quantitative methods for cell-shape analysis. *Journal of Microscopy*, 227(2):140–156, August 2007.

[40] Susanne M. Rafelski and Julie A. Theriot. CRAWLING TOWARD A UNIFIED MODEL OF CELL MOTILITY: Spatial and Temporal Regulation of Actin Dynamics. *Annual Review of Biochemistry*, 73(1):209–239, 2004.

[41] Nilanjan Ray and Scott T. Acton. Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *IEEE transactions on medical imaging*, 23(12):1466–1478, December 2004.

[42] A. Sacan, H. Ferhatosmanoglu, and H. Coskun. Celltrack: an open-source software for cell tracking and motility analysis. `http://bio.cse.ohio-state.edu/CellTrack/`. Database Research Group, Ohio University.

[43] A. Sacan, H. Ferhatosmanoglu, and H. Coskun. Celltrack: an open-source software for cell tracking and motility analysis. *Bioinformatics*, (14):1647–1649, 2008.

[44] Ahmet Sacan, Hakan Ferhatosmanoglu, and Huseyin Coskun. CellTrack: an open-source software for cell tracking and motility analysis. *Bioinformatics*, 24(14):1647–1649, July 2008.

[45] E. D. Salmon and Phong Tran. High-Resolution Video-Enhanced Differential Inter-ference Contrast Light Microscopy. In Greenfield Sluder and David E. Wolf, editors, *Digital Microscopy*, volume 81 of *Methods in Cell Biology*, pages 335 – 363. Aca-demic Press, San Diego, California, third edition, 2007.

[46] Javier Satulovsky, Roger Lui, and Yu-li Wang. Exploring the Control Circuit of Cell Migration by Mathematical Modeling. *Biophys J*, 94(9):3671–3683, May 2008.

[47] S Schwartz, D.B. Murphy, K.R. Spring, and M.W. Davidson. de sénarmont bias retardation in dic microscopy. `http://www.microscopyu.com/pdfs/DICMicroscopy.pdf`, 2003. Nikon MicroscopyU.

[48] I. Simon, C. R. Pound, A. W. Partin, J. Q. Clemens, and W. A. Christens-Barry. Automated image analysis system for detecting boundaries of live prostate cancer cells. *Cytometry*, 31(4):287–294, April 1998.

[49] N. M. Sirakov, H. Kojouharov, and N. N. Sirakova. Tracking neutrophil cells by active contours with coherence and boundary improvement filter. In *Image Analysis & Interpretation (SSIAI), 2010 IEEE Southwest Symposium on*, pages 5–8. IEEE, May 2010.

[50] D.R. Soll and D. Wessels. *Motion Analysis of Living Cells*. Techniques in Modern Biomedical Microscopy. Wiley, 1997.

[51] J. R. Soong. A549 bridges. `http://commons.wikimedia.org/wiki/File:Intercellular_connections_in_a549_cells.jpg`, August 2010. [Online] Accessed 01 December 2014.

[52] David J. Stephens and Victoria J. Allan. Light Microscopy Techniques for Live Cell Imaging. *Science*, 300:82 – 86, April 2003.

[53] Aaron F. Straight, Amy Cheung, John Limouze, Irene Chen, Nick J. Westwood, James R. Sellers, and Timothy J. Mitchison. Dissecting Temporal and Spatial Control of Cytokinesis with a Myosin II Inhibitor. *Science*, 299(5613):1743–1747, March 2003.

[54] D. Tomazevic, B. Likar, and F. Pernus. Comparative evaluation of retrospective shading correction methods. *Journal of microscopy*, 208(Pt 3):212–223, December 2002.

[55] Luminita A. Vese and Tony F. Chan. A Multiphase Level Set Framework for Image Segmentation Using the Mumford and Shah Model. *International Journal of Computer Vision*, 50(3):271–293, December 2002.

[56] G. Wenzel. Filopodia detection and measurement for normal human epidermal keratinocytes on different substrata. Diploma thesis, Bonn University, 2003.

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

[58] D. Young and A. J. Gray. Cell identification in Differential Interference Contrast microscope images using edge detection. pages 66.1–66.10. 7th British Machine Vision Conference, BMVA Press, 1996.