Algorithms for automated image analysis of *Schizosaccharomyces pombe* cells viewed under phase-contrast/brightfield microscopy

Jennifer Anne O'Brien

A thesis presented for the degree of

Doctor of Philosophy

School of Engineering and Digital Arts

University of Kent

UK

September 2018

180 pages

Abstract

Schizosaccharomyces pombe (*S. pombe*) are used as model organisms to discover intracellular interactions and if mutations have the potential to affect higher eukaryotes, such as humans. Due to their regimented growth pattern, transmitted light microscopy can be used to identify abnormalities in cells, often by simple measures such as cell length and width.

Algorithms for automated image analysis of microscopy images assist biologists by providing consistent, comparable results while reducing the demand on researchers' time. This thesis introduces a novel algorithm for making ground truth datasets, a novel algorithm for segmenting *S. pombe* cells from phase contrast or transmitted light microscopy images, and an algorithm to measure the lengths and widths of the segmented cells.

The novel manual segmentation algorithm allows for accurate and precise, form-fitting ground truth segmentations unlike currently available programs. The novel segmentation algorithm functions better than PombeX (the current existing program designed for the same problem) on both phase contrast and transmitted light microscopy images of *S. pombe*. The measurement-making algorithm produces length and width measurements that closely match manual measurements.

Declaration

I declare that this thesis is an original report and all the work is my own, except where stated otherwise by reference or acknowledgement.

No part of this thesis has been submitted in support of an application for any degree or qualification from the University of Kent or any other university or institute of learning.

Jennifer O'Brien

Acknowledgements

I would like to thank my supervisors, Drs Daniel Mulvihill, Konstantinos Sirlantzis and Sanaul Hoque, for their advice, knowledge and support throughout this PhD.

Thank you to the Industrial Biotechnology Centre for providing my funding in the form of a University of Kent 50th Anniversary Research Scholarship.

A big thank you to the support services in both the School of Biosciences and the School of Engineering and Digital Arts for all their help over the years.

I would like to thank my colleagues in the Intelligent Interactions group and MaDCaP group for their help and friendship. Especially those in MaDCaP who lent their expertise for determining the variance in manual segmentation of cells and Irene Gyamfi for supplying an independently collected dataset.

Finally, I have to thank my friends and family for their unwavering encouragement and support.

Table of Contents

List of Tables

2 Materials and Methods

3 Ground Truth Segmentations

4 A Novel Segmentation Algorithm for Phase Contrast and Transmitted Light Microscopy Images of *Schizosaccharomyces pombe*

5 Measuring Cell Lengths and Widths

List of Figures

1 Introduction

2 Materials and Methods

3 Ground Truth Segmentations

4 A Novel Segmentation Algorithm for Phase Contrast and Transmitted Light Microscopy Images of *Schizosaccharomyces pombe*

5 Measuring Cell Lengths and Widths

List of Abbreviations

°C	degrees Celsius
$\geq$	greater than or equal to
$<$	less than
$\leq$	less than or equal to
µm	micrometre
µm^2	micrometre squared
%	per cent
±	plus or minus
2D	two dimensional
3D	three dimensional
B-spline	basic/basis spline
c value	constant used for determining the second dilation radius
CCD	charged coupled device
DNA	deoxyribonucleic acid
EMCCD	electron multiplication charge coupled device
EMM	Edinburgh minimal media
fn	false negative
fp	false positive

G segments	ground truth segments
G1 phase	gap 1 phase
G2 phase	gap 2 phase
GCE	global consistency error
GFP	green fluorescent protein
GVF	gradient vector flow
J value	Jaccard index value
LCE	local consistency error
LEDs	light-emitting diodes
LoG	laplacian of gaussian
LRE	local refinement error
M phase	mitosis phase
mag	magnification
MEM	Martin error measure
MIT	Massachusetts Institute of Technology
NPR	normalised probabilistic Rand
OCE	object-level consistency error
p	precision
P segments	proposed segments
PEM	partial error measure
PS	proposed segments
r	recall
R index	Rand index
rRNA	ribosomal ribonucleic acid
S phase	synthesis phase
S. pombe	*Schizosaccharomyces pombe*

sCMOS	scientific complementary metal oxide semiconductor
tp	true positive
UV	ultraviolet
YFP	yellow fluorescent protein

CHAPTER 1

Introduction

Humans have a vested interest in understanding how cells work as it extends our basic understanding of biological processes; and this knowledge leads to treatments and cures for diseases and illnesses, and even things like *in vitro* fertilisation. Biological studies are used to test hypotheses on a range of topics including intracellular transportation, DNA repair, and cell communication to name a few.

1.1 History of Cells and Organisms

Cells are the building blocks of life. The countless different cell types currently found on Earth are descendants of the first cell(s) which originated billions of years ago. Organisms can be unicellular or multicellular. Multicellular organisms consist of differentiated cells that work together to ensure the survival of the organism – the individual cells or component systems of a plant or animal (for example leaves, roots, bones, or organs) cannot live without assistance from the rest of the organism.

Genetically, cells can be divided into three domains: archaea, bacteria, and eukarya [1]. While it is impossible to truly identify the original cell line, the most recent common ancestor for the domains has been determined [1, 2]. The organisms within each domain have characteristic features, most notably the type of rRNA contained in the ribosomes [1]. The evolutionary relationship between organisms can be represented by a phylogenetic tree (figure 1.1).

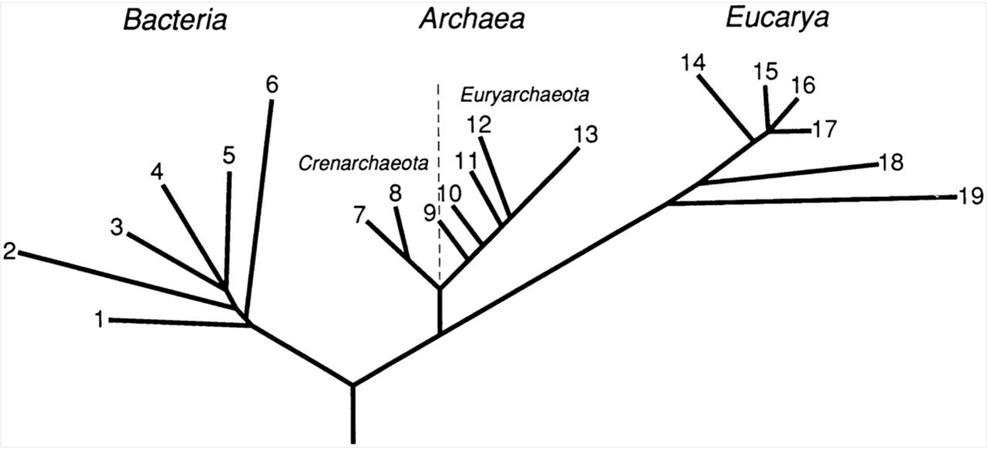


Figure 1.1: Phylogenetic tree representing the evolutionary distance between domains and organisms. Taken from [1].

The archaea domain consists of prokaryotic (have no membrane-bound nucleus [3]) cells [1] and are often considered to be extremophiles because they thrive in traditionally inhospitable environments [4]. Organisms of the bacteria domain are prokaryotes [1]. Bacteria are found across the world and are widely studied as they are involved with many facets of everyday human life. Cells belonging to the eukarya domain are eukaryotic (have a membrane-bound nucleus and other membrane-bound organelles [3]) [1]. The domain can be split into five kingdoms: protozoa, chromista, plantae, animalia (which includes humans, branch 14 in figure 1.1), and fungi (which includes yeasts, branch 17 in figure 1.1) [5].

1.2 Eukaryotic Cell Growth

Eukaryotic cells grow and replicate via a well defined cell cycle. The cycle is split into four stages: gap 1 (G1) phase, synthesis (S) phase, gap 2 (G2) phase, and mitosis (M) phase (figure 1.2) [6]. During the gap phases cells grow in size and components required for two cells are synthesised. There is a checkpoint in each gap phase to ensure the cell is prepared to divide. DNA is replicated during S phase and the M phase is where the cell organises the DNA in the nucleus, and cytokinesis occurs. The M phase is also where the final checkpoint is before the cell divides, however, the cell is committed to the cell cycle once it passes the checkpoint in G1 phase [7]. The amount of time spent in each phase and the total length of time for one complete cell cycle varies between cell types [6].

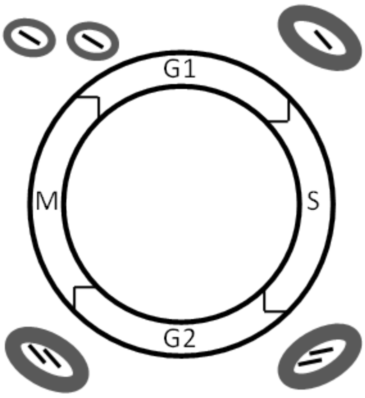


Figure 1.2: Representation of a cell (grey) and its nucleus (white, with DNA in black) during the eukaryotic cell cycle. G1 is the first gap phase, S is the synthesis phase, G2 is the second gap phase, and M is the mitotic phase which ends with cytokinesis.

Homeostasis is achieved when internal conditions are maintained at desirable levels, irrespective of external factors. This is achieved by the use of alternate (stress) pathways to adapt to conditions; however, cell growth and function can be affected by the use of these different pathways [8, 9, 10, 11]. Stress pathways can be activated in response to extracellular stressors such as a lack of nutrients, unfavourable gaseous environments, or unsuitable temperatures.

To ensure 'normal' growth for studies, cell cultures are grown and maintained in log phase [12] (figure 1.3). Log phase is considered optimal for studies as during lag phase cells are at extremely low densities and considered inactive. During log phase the cells are increasing exponentially in number, which demonstrates that the cells are dividing at a steady rate. Once cell cultures reach the stationary phase, there are limiting factors such as nutrient availability, which results in cells exiting the cell cycle and eventually cell apoptosis exceeding cell proliferation; ultimately causing the death phase [13, 14, 15]. This means that homeostasis has failed. Concentrating cells for microscopy studies can be achieved by centrifugation of the culture, however, this stresses the cell [16].

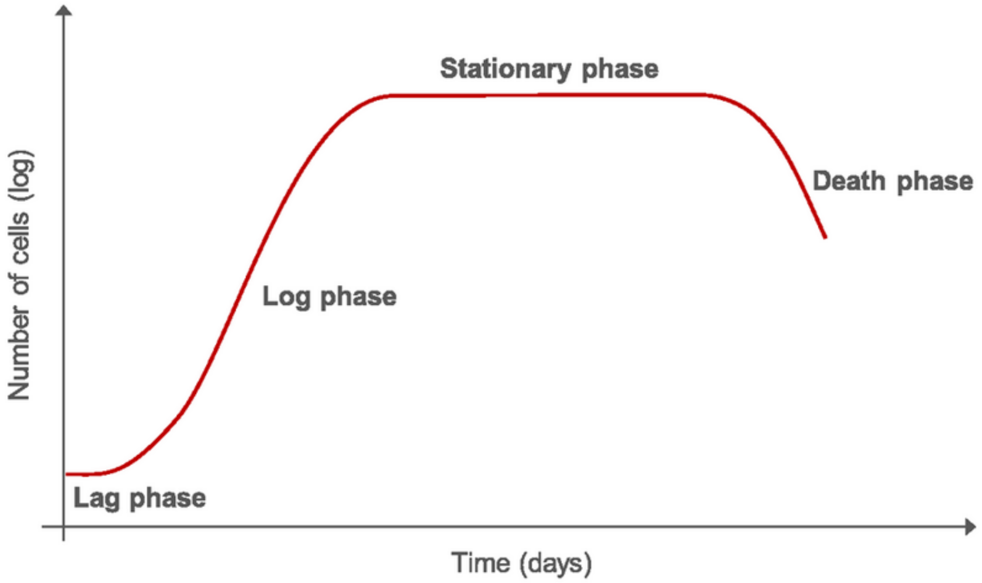


Figure 1.3: Typical growth curve for cells with monoauxic growth. Taken from [17].

1.3 Genetic Manipulation of Cells

Genetic information is stored in the nucleus of eukaryotic cells on strands of DNA. This DNA holds the information needed to synthesise every protein found in the organism, as well as redundant sections. Mutations in DNA can cause illness and/or disease and therefore the basis of many biological studies.

1.3.1 Mutant cells

Mutations can be inherited from parent cells/organisms, resulting in hereditary diseases like cystic fibrosis, or acquired – either by replication errors during S phase or by DNA damaging agents such as UV light [18, 19] – which can lead to diseases such as cancer if not repaired. Mutations can also be purposefully introduced in a controlled, known manner via site-directed mutagenesis [20].

There are several protocols for site-directed mutagenesis [20, 21, 22]. It requires previous knowledge of the sequence of the gene to determine the target. The overall mechanism is simple and universal: a synthetic DNA primer with the desired mutation is used to 'overwrite' the original gene sequence during DNA replication [20]. This can be done in a plasmid (a circular piece of DNA that can be inserted into cells and integrated with chromosomes) or *in vivo.*

Cells which have been subjected to mutagenesis attempts need to be screened to ensure that the desired mutation has been incorporated. This can be done in several ways, however, it is crucial that each potential cell line is assessed individually and cultures grown from one singular cell. Screening can be done inserting an additional, non critical mutation, for example by modifying an enzyme cut site and assessing the fragments of enzymatically cut DNA.

1.3.2 Fluorescent proteins

Green fluorescent protein was first discovered in jellyfish and their discovery was awarded the 2008 Nobel Prize for Chemistry [23]. Since the discovery of green fluorescent protein, there has been a boom in the development of more fluorescent proteins [24, 25]. Fluorescence occurs when electrons in the fluorescent protein are excited by light and then revert to their ground state by releasing a photon. Each fluorescent protein has a specific excitation and emission wavelength associated with it (figure 1.4) and this determines the colour of the emitted light seen for each protein.

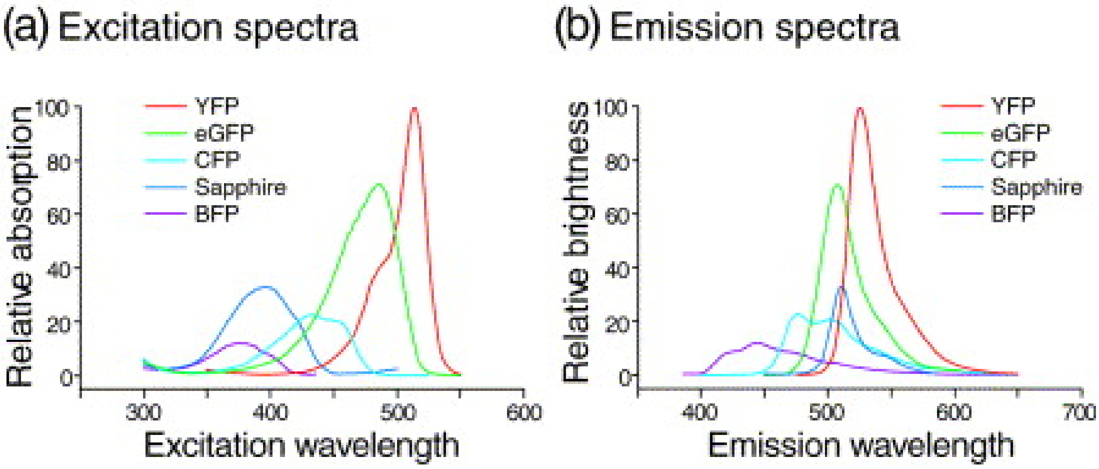


Figure 1.4: The (a) excitation and (b) emission spectra of some fluorescent proteins (relative to yellow fluorescent protein (YFP) = 100%). Taken from [26].

The gene encoding a fluorescent protein can be spliced to a protein-encoding gene. This causes the targeted protein to be synthesised with the fluorescent protein covalently attached to the end of a protein, allowing it to be observed *in vivo* [25]. As with all mutations, the ability of the protein to fold correctly can be affected and this effects protein function. To ensure that this does not occur, a copy of the mutant cell line without the fluorescent protein should be constructed and observed to ensure no differences, either in phenotype or growth pattern. Changes indicate altered function caused by the addition of the fluorescent protein.

Studies can be designed to use multiple fluorescent proteins [26]. The excitation and emission wavelengths for each fluorescent protein to be used need to be considered. If the excitation wavelengths for the proteins overlap, the emission wavelengths need to be distinct to avoid the excitation wavelengths being registered as emission wavelengths and creating bleed-through. This risk is reduced by filtering both the excitation and emission wavelengths and only activating and imaging one excitation and one emission wavelength at a time.

1.4 Model Organisms

While the end goal of most biological studies relates to human-based problems, non-human and even non-animal cells are used in studies. There are several reasons why different organisms are sometimes preferred for studies, just as there are factors to consider when selecting suitable substitute organisms.

1.4.1 Why use model organisms

Model organisms are possible because many pathways – from the cell cycle to respiration – have homologous proteins across the eukarya domain. Due to isoforms (functionally similar proteins with different primary structures), homologs are not always linearly aligned. Multicellular organisms tend to have more isoforms than unicellular organisms due to the differentiation of cells. This means that there are more genes encoding a group/family of proteins, each of which could be responsible for causing illness/disease. Using organisms with fewer individual isoforms but homologs to each isoform means that knowledge about one protein in a less complex organism can provide information for several homologous isoforms in higher eukaryotes.

The second reason model organisms are considered is because less complex organisms are easier to grow, replicate faster, and are easier to manipulate genetically. This means it is possible to use them to screen potential mutations in order to isolate specific mutations which affect cell growth/function. Further studies can then be used to investigate these promising mutations in more complex systems. It is also easier to ensure cultures of unicellular organisms maintain their desired mutations [27].

1.4.2 Selecting model organisms

The most important factor to consider when selecting a model organism is that the organism supports the study's aims. This means that the protein(s)/pathway(s) must be conserved between the model organism and the organism for which the information is wanted. Other factors to consider are if the organism is a step towards the final organism of interest or if tissue cultures (cultures of differentiated cells) would be more suitable for the next cell to study.

1.4.3 *S. pombe* as a model organism

Schizosaccharomyces pombe (*S. pombe*) are rod-shaped, unicellular eukaryotic yeast cells. They have a doubling time of anywhere between 2 and 10 hours depending on media and temperature [16]. For most studies, suggested densities range from 1×10^6-5×10^6 in simple media to 1×10^6-1×10^7 in complex media [16]. The effects of media and temperature on the growth of *S. pombe* cells means that it is important that conditions remain consistent throughout a study.

The growth of wild type *S. pombe* cells is well-defined. Haploid cells grow via tip extension, maintaining a diameter of approximately 3.5 µm [16, 28], to a maximum length of 12-14µm [16, 29] before dividing. Once critical mass is reached, the cell divides. The ability of *S. pombe* to survive as haploid cells is beneficial as it allows for straightforward studying of recessive genes [27].

The *S. pombe* genome was sequenced in 2002 [30] and has since been annotated. According to PomBase [31], there are a total of 5138 protein-encoding genes with 149 proteins that are specific to *S. pombe* alone and 4540 proteins that are conserved in eukaryotes; therefore, genetically, *S. pombe* are suitable model organisms for preliminary experiments.

Site-directed mutagenesis is easier in *S. pombe* compared to human cells for several reasons. First, they can support plasmids [27] which are components of several of the site-directed mutagenesis protocols. Secondly, *S. pombe* have a higher occurrence of recombination [27]. This is key again is it is how the plasmid's DNA is integrated into the cell's DNA.

Structurally, the contents of *S. pombe* cells are similar to those of mammalian cells. Besides the membrane-bound nucleus, other organelles/structures the cells have in common include: vacuoles, lysosome-like vacuoles [32], mitochondria, ribosomes, Golgi apparatus, and endoplasmic reticulum [28, 33]. These organelles indicate that the same metabolic pathways are used by *S. pombe* and mammalian cells, and this is further strengthened by the homologous proteins.

The largest issue with *S. pombe* cells being used as a model organism for mammalian cells is that they have cell walls [28] while animal cells do not. This is only problematic in studies involving extracellular interactions where the cell wall may be involved. For studies which only concern intracellular activities, *S. pombe* are ideal model organisms.

1.5 Microscopes and Microscopy

Microscopy is commonly used in biology to examine microorganisms. Since ancient times, lenses have been used to magnify objects which are too small to be visible to the naked eye. The technology and capability of microscopes has grown and this has led to numerous types of microscopy, each of which has various characteristics that determine their usefulness to a study.

1.5.1 Types of microscopy

The intrinsic principle of microscopy is to magnify objects so that the fine structures can be observed. There are several mechanisms that can be used to enhance images from using optics (optical microscopy), to detecting structures with physical probes (scanning probe microscopy). Sound waves, x-rays and electron beams can also be used to visualise fine details. Each method has its own strengths and weaknesses; however, for the purposes of microbiology, optical microscopy is the most commonly used method [25, 34, 35, 36, 37]. This is because it is comparatively cheap, relatively simple, and highly effective. Optical microscopy also offers the benefit of being able to image living cells, which is important for many studies and crucial for time lapse imaging.

1.5.2 Optical microscopy

1.5.2.1 System set up

Optical microscopy uses a light source and magnifying lenses to allow the visualisation of objects. The light illuminates the object and the image is then passed through a series of lenses to magnify it and allow it to be viewed (figure 1.5).

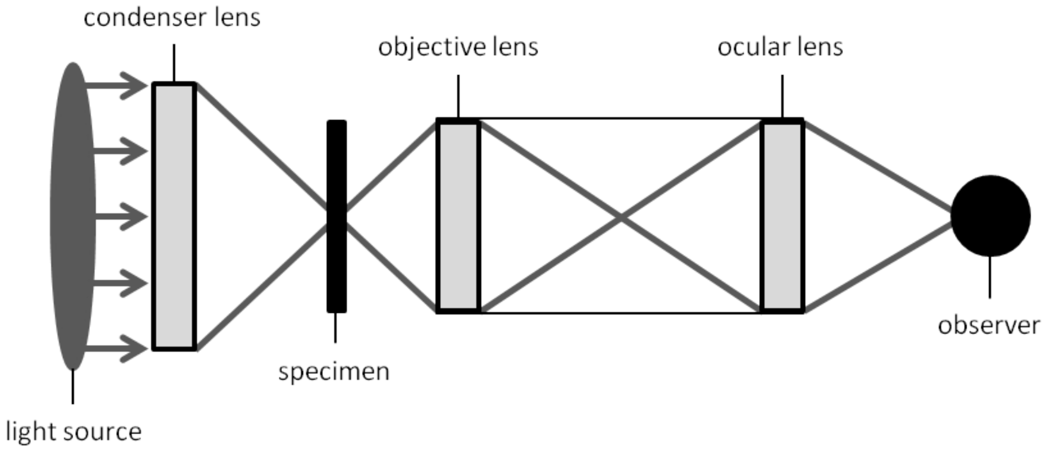


Figure 1.5: Schematic of a light microscope.

1.5.2.2 Light source

Microscopes can use light from a number of sources, with the most common being a lamp. These lamps can have a variety of internal constructions, from tungsten-halogen lamps to xenon arc lamps. Lamps are the often cheaper and easier to set up, however, the maximum luminescence is not instantly reached so a consistent, steady light source is not instantaneous. Controlling the quantity and distribution of illumination between users is difficult with traditional lamp light sources but condenser lenses help with this (figure 1.5).

Light-emitting diodes (LEDs) are a longer lasting, more robust light source when compared to lamps, but can be more expensive. They have a negligible timeframe for reaching maximum luminescence and both the number of, and position of, LEDs that are on at any point can be controlled. This reduces the risk of photobleaching, especially in time lapse sequences, and also allows for greater control over the overall brightness.

1.5.2.3 Microscope lenses

All microscope lenses have a magnification factor. The total magnification of an object is the product of the magnification for all lenses between the slide and the observer [38]. The higher the total magnification, the larger the cells appear. Over-magnification rarely occurs, however, the level of magnification should be considered to ensure that it gives sufficient detail without losing focus [38].

Lenses can also be suited for a specific immersion medium between the objective lens and the slide. Immersion media reduce the differences between the refractive indices of each part of the imaging system compared to air as an intermediary [39]. This reduces the diffraction of light (figure 1.6) and prevents the loss of information.

Numerical aperture (NA) defines the maximum resolution for the image seen by the observer [40]. It is the area from which light is focussed by the objective

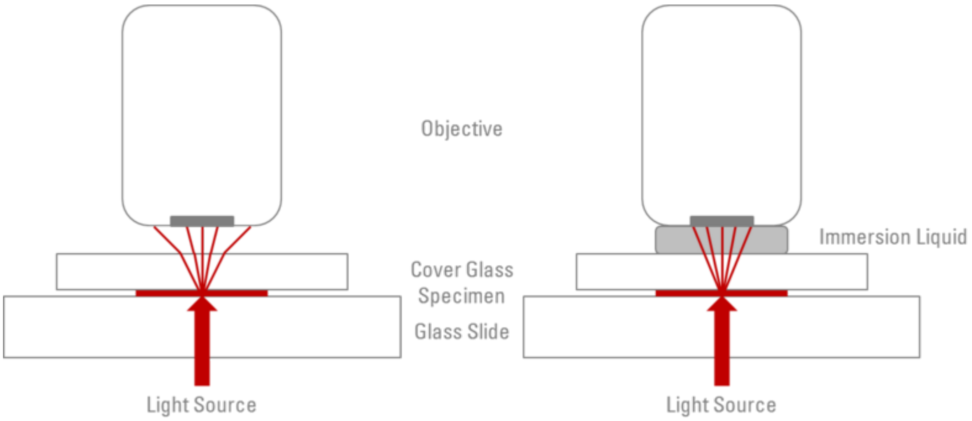


Figure 1.6: Diagram showing how the pathway light takes varies based on the refractive index of the media it passes through. The pathway when the refractive index of the immersion liquid is (L) lower than that of glass or (R) the same as glass. Taken from [39].

lens to produce the end image at a fixed distance. Rays of light outside this area do not contribute to the image. The larger the NA, the higher the resolution of the image. It is calculated by equation 1.1 where n is the refractive index of the medium between the objective lens and specimen and α is half of the objective aperture.

$$NA = n \cdot sin(\alpha) \tag{1.1}$$

1.5.3 Types of optical microscopy

There are many types of optical microscopy which are used in biological studies. Brightfield, phase contrast, and fluorescence microscopy are all used in *S. pombe* studies [41, 42, 43]. These three are the only types used in this thesis.

1.5.3.1 Brightfield/transmitted light microscopy

Brightfield, or transmitted light, microscopy is the most basic type of microscopy. It uses the full spectrum of light and high magnifications to visualise transparent objects [38], which includes most cells. The resultant images are true in colour, meaning various stains can be used to highlight specific structures/regions. These stains are often toxic to the cell though, and therefore may not ideal for some biological studies.

1.5.3.2 Phase contrast microscopy

The development of phase contrast microscopy won the 1953 Nobel Prize in Physics [44]. It is similar to brightfield microscopy as the details visualised occur due to different refractive indices within the specimen producing different shifts in the wavelengths which pass through them [38]. Phase contrast microscopy images have enhanced contrast compared to brightfield microscopy images which allows for detailed internal structures to be seen in living cells.

Maximum contrast is achieved by inserting an annular ring (figure 1.7a) over the light source, and a phase plate (figure 1.7b) between the objective lens and the ocular lens. The annular ring allows a cone of light to illuminate the specimen and any illuminating beams that do not pass through the sample are blocked by the opaque ring on the phase plate. Light that passes through the specimen is undergoes a phase shift and is diffracted. This allows the waves to pass through the transparent portion of the phase plate where the phase is again retarded. The light beams are then summed, and the destructive interference of the maximally 180° phase shifted beams result in black regions [38].

(a) Annular ring

(b) Phase plate

Figure 1.7: Illustration of annular ring and phase plate.

While it is beneficial that the additional contrast produces more detailed images without the use of potentially toxic stains, phase contrast microscopy has its own issues. The major problem arising from phase contrast microscopy is the loss of true colour image capabilities. This means that only differences in density within the object can be observed. False colouring of the observable images can be achieved by using colour filters [45]; however, this provides no additional data.

1.5.3.3 Fluorescence microscopy

Fluorescence microscopy makes use of intrinsic and/or extrinsic fluorescence. The principle behind it is that fluorophores absorb and emit at known wavelengths [24]. Both the incidental and transmitted light are filtered by wavelength, as defined by the fluorescence spectra for the targeted fluorophore. By controlling the wavelength of the incidental light, multiple fluorophores can be used within the same specimen. This is utilised in studies such as those that look at how molecules interact with each other *in vivo*. Consideration of the fluorescence spectra of multiple fluorophores is key and there is a limit to the number of fluorophores that can be used at any one time [35].

1.5.4 Mounting samples

For all types of optical microscopy, the sample needs to be mounted on a slide. Often, the mounting of cells for biological studies needs to immobilise them to allow for clear images over long periods of time (time-lapse microscopy) to be taken for further analysis. There are many different methods to adhere cells to various types of slides; each method can affect things such as the quality of the images and the activity of the cells.

1.5.4.1 Slide selection

Imaging cells can take varying amounts of time. This means that there is a risk of cells activating stress pathways or dying, both of which can skew experimental results (as discussed in section 1.2). This can be avoided by providing the cells with growth media and maintaining the temperature required by the cells for optimal growth. Studies which use only single-time-point images do not require much media as the necessary images are acquired reasonably fast after the cells have been mounted. This is also true for time lapse studies which span short periods of time.

Long-period time lapse studies require a larger supply of growth media to ensure homeostatic growth of cells. This means traditional slides are often unsuitable as they cannot hold the necessary volume of liquid. Imaging chambers can be used instead. The watertight spacers used between the slide and the slide cover allow for the growth media to be contained and therefore ensure homeostasis is maintained.

1.5.4.2 Methods to adhere cells to the slide

Cells which are adherent in nature (for example cells from mammalian tissue samples) can be grown directly on sterile coverslips, ready for imaging. However, cells adhere poorly to glass coverslips (which are necessary for high quality imaging) so reagents have been developed to help cells adhere [46, 47]. This method is not ideal for unicellular organisms, such as *S. pombe*, as they will not grow in a single layer, but will form clusters if not kept shaken. If they are shaken, the cells will not settle onto the coverslip and therefore nothing will adhere.

It is important that the cells are kept in log phase from mounting to the end of imaging, unless the study indicates otherwise, so mounting cells from liquid cultures is ideal. As *S. pombe* are not naturally adherent, a ‘glue’ is required to immobilise the cells to ensure clear images can be captured. The reagents used as glue have to be carefully selected to ensure they do not interfere with the imaging, however, there are numerous options available, with lectin being commonly used [48].

Selecting suitable adhering methods means that intrinsically fluorescent substances cannot be used in fluorescence microscopy (unless the fluorescence is to be used to determine the outside of the cells). It also means that reagents which put the cells into stasis cannot be used for time lapse studies. Usually, each laboratory uses the method(s) set by the primary investigator, based on their experiences, their microscope set up, and the type of study being conducted.

1.6 Imaging Hardware and Software

1.6.1 Imaging Hardware

Viewing the images can be done either visually or with a camera in microscopy. For biological studies, it is important that results are reproducible and this means that capturing the images allows for computer assistance in the analysis of data, as well as producing example images. This means that cameras are commonly used.

Camera selection for a study is based on the best camera available in the laboratory as they are expensive pieces of equipment. However, there are a number of things which are important to consider when selecting a camera for a study: resolution, chip size, and type of chip.

1.6.1.1 Resolution

The resolution is the physical area represented per pixel. It is important in microscopy as it determines the level of detail that can be captured and the overall clarity of the image. The significance of resolution is affected by the magnification of the microscope. This is because the higher the magnification, the more detailed the image and, therefore, the loss of information from lower resolution cameras is lessened. Microscope magnification has limitations as to how far it can overcome resolution, however, it is important to consider when choosing between cameras.

1.6.1.2 Size of camera chip

The size of the chip used determines the resolution the image is saved at. This can affect the level of detail visible when viewed as the frame size is limited by the viewing screen size. Inaccuracies can arise when manual analysis is carried out on these images if zoom functions are not used, but automated analysis is not affected so long as the files are not compressed when saved.

1.6.1.3 Type of camera chip

Camera chips are the digital version of film. Analogue cameras have shutter speeds that determine the length of time the film is exposed to capture the image. The quality of the image varies depending on the quality of the subject's illumination and shutter speed. Digitally, shutter speed refers to the speed at which the data from each pixel can be read, stored, and reset [49].

There are several different types of sensors which can be used to capture an image. Charged coupled device (CCD), electron multiplication charged coupled device (EMCCD), or scientific complementary metal oxide semiconductor (sCMOS) chips are commonly used in microscopy [49].

CCD chips transfer data at the slowest rate of the three chip types with only one output channel [49], and therefore, are unsuited to clearly capture images with low contrast levels. To improve the capture of dim images (such as those seen in fluorescence microscopy) EMCCD chips are more suitable. They amplify the light that reaches them, increasing the level of contrast in the image [49]. sCMOS chips have multiple output channels which increases the rate of data transfer [49]. This improves the capture of images in dim light.

1.6.2 Imaging software

Computer software is used to control many aspects of microscopy imaging. Everything, from switching between brightfield/phase contrast microscopy and fluorescence microscopy to the exposure time given for the camera, can be managed by imaging software. It can also provide directions for time lapse experiments, instructions for multiple z-stack images, and the software can be linked to analysis algorithms.

There are countless programs available for managing microscopy imaging [50, 51, 52]. Some of these programs excel in all fields, while others are targeted to a specific subset of functions. It is usual for the imaging software available in a laboratory to be chosen based on the primary investigator's own experiences.

1.7 Image Analysis Overview

Image analysis is a crucial stage of any microscopy-based study. It isolates, or segments, regions of interest and provides information about many things: any phenotypical differences between strains, where targeted molecules are located, how targeted molecules move (speed, patterns, etc), whether targeted molecules interact with each other, as well as other such things. Without analysis, microscopy does not add to a study.

Before cameras and computers were capable for analysis, all image analysis was done manually with a researcher taking measurements and sketching while sat in front of the microscope. Fully manual analysis is susceptible to numerous errors and highly subjective. As human error is the main perpetrator of these problems, minimising human participation is ideal. The introduction of, and advances in, both computers and cameras has made this possible; using either a semi-automated or a fully automated system which increases efficiency, consistency, and completeness in studies [53].

There are different levels of semi-automated analysis depending on the amount of input the end-user has. This can range from extremely labour intensive, with almost every step for every frame needing input, to systems that require some initial input parameters for a dataset as a whole. Image analysis systems can never be truly fully automated because some end-user participation is always required, even if it is just to select the algorithm and frames for analysis.

In this thesis, any algorithm which requires the end-user to use their personal judgement to provide input data is considered a semi-automated system, and all others are considered automated. This demarcation was chosen as personal judgements are dependent on subject-specific knowledge, and therefore, subjective.

1.7.1 Semi-automated image analysis algorithms

The range of user input for semi-automated image analysis algorithms varies. The least user friendly type of semi-automated software are those like ImageJ [54] and CellProfiler [55]. Both of these rely on the user to choose and order a series of functions to create a ‘pipeline’. While there are numerous existing pipelines that can be downloaded and modified, it can be difficult for non-programmers to develop an informative pipeline. The main benefit of both of these programs is that they are freely available and do not require commercially available software to run.

More intuitive algorithms rely on parameter selection by the user rather than the construction of the pipeline. Often, the input information is necessary as the algorithm is not specific for an organism, and therefore, prior knowledge needs to be provided. However, it also means that mutations which cause abnormal phenotypes are segmentable as any biases established are on a dataset-by-dataset basis. One example is CellX [56].

CellX [56] is a program designed to segment any type of cell from transmitted light, phase contrast, or fluorescence microscopy by relying on the contrast between the edge of the cell and the background. It requires interactive input to set the seed size, maximum cell length, and membrane pattern (the contrast gradient, membrane width, and location) [56, 57]. While the cell length and seed size could be pre-set if the images are of the same cell type, the membrane pattern varies between imaging series depending on microscope set up meaning that it has to be set for each dataset individually. Furthermore, the parameters are determined using one image. This means that the maximum cell length is susceptible to being dramatically incorrect if the image selected shows only the smallest cells.

1.7.2 Automated image analysis algorithms

Fluorescence microscopy lends itself to automated image analysis due to the stark contrast between regions of interest and the background. The levels of noise in transmitted light/phase contrast microscopy means that it is more difficult to determine the true edge of a cell. This has led to a large number of segmentation algorithms for fluorescence microscopy [58, 59, 60], but a limited number of segmentation algorithms for transmitted light/phase contrast microscopy [61, 62], which are often organism (or cell shape) specific. Some automated algorithms can rely on additional fluorescence microscopy images to improve their segmentation success of transmitted light/phase contrast microscopy images, for example PombeX [62].

The amount of effort the user needs to expend for automated algorithms varies; however, the work can be done irrespective of knowledge. For example, every image has to be selected individually for segmentation by PombeX [62], which is labour-intensive but does not require expertise. On the other hand, CellSegm [58] can process entire datasets with one command; however, it provides several algorithms of which the end-user must select the one suited to the type of analysis wanted (eg 2D or 3D). This is not labour-intensive but requires some expert knowledge as to the focus of the study.

Overall, the most important feature of automated image analysis algorithms is their ability to objectively segment regions of interest. It is important that the algorithm is able to process abnormal phenotypes within reason so that a wide array of mutant strains can be studied. To ensure consistent, objective segmentation, parameters need to be set independently of the dataset.

1.8 Techniques Used in Image Analysis Algorithms

1.8.1 Neural networks and machine learning

Neural networks and machine learning techniques have been popular mechanisms for image segmentation over the last decade [63, 64, 65, 66]. A neural network can be used to determine where cells are; however, determining all the neurons needed to describe a cell is difficult to do without creating bias, especially as the cells have random orientation. This problem is also faced by machine learning algorithms. It is difficult to gain enough annotated data for the learning component as well as the fact that not all phenotypes can be 'taught' to the system. This means only cells of a 'typical' phenotype will be segmented and cells of interest will be missed.

While other segmentation methods may be biased towards targeting stereotypical cells, it is possible to identify the cells that would be excluded. The smallest/largest potential segments can be calculated with knowledge of the algorithm's parameters, and cells with phenotypes that lay outside of these segments are excluded. The reason this is not possible for neural networks is because of the large number of neurons that need to be defined for success limits the amount of variability in segmented regions. Machine learning relies on input images so any phenotype which has not registered in the system cannot be identified in the subsequent images if it differs too much from the norm.

1.8.2 Traditional image analysis methods

Image segmentation algorithms are built from techniques that fall into four main groups. These groups are: image enhancement, morphological filtering, image segmentation, and region analysis [67]. Each of these groups contain numerous algorithms and functions, and for every problem a different sequence of these is optimal.

Image enhancement is responsible for reducing noise and increasing contrast in the image. While an image can appear clear visually, it is important to remember that computers do not ‘see’ in the same way as humans. Images to a computer are an array of matrices which denote the intensities of each pixel in each colour plane. This means that while images may not appear altered through enhancement, it greatly improve a segmentation algorithm’s success.

Morphological filters can be used to enhance specific features within the image. Possible functions include filling regions, eroding/dilating regions/lines, filtering by area size, or detecting features (such as edges) [53, 67]. The end result of this filtering is to further reduce noise and emphasise regions of interest. The enhancement of images and the use of morphological filters reduces noise, however, it also compromises the overall resolution of the image [53]. This means that there is a practical limit to the amount of manipulation the image can undergo while still being useful.

Image segmentation is the most commonly desired functions of automated image analysis because identifying and isolating regions of interest is the most time consuming and subjective part of image analysis. Until the regions have been segmented, the enhancement of the image is unproductive.

Region analysis is the final stage of any segmentation algorithm. It is where the geometric measurements are made and statistics run. There are many functions in MATLAB [68] which are capable of this, with regionprops being the simplest. Outside of these functions, there are also the semi-automated analysis algorithms mentioned in chapter 1.7.1.

1.9 Considerations for Segmenting *S. pombe* Cells

It is important to consider the specific nature of the problem at hand when choosing or designing image analysis algorithms. There are countless factors that are unique for each scenario depending on the type of images to be analysed. For example, the identification of road signs in full colour images relies on matching potential regions of interest to example images [69]. This means that colour filters can be used during the segmentation stages and the segmentations need not be 'true fit' as the possible matching outcomes do not have to be perfect matches. Algorithms which fit these criteria would perform poorly on microscopy images as colour filters have no effect.

The second major consideration for analysing *S. pombe* cells is that segmentation leads to geometric measurements, and therefore, the segmentation of regions must be 'true fit'. This is complicated by cells that are touching in images that ought to be segmented as individual cells, furthermore, cells which have just divided or about to divide are subjective to the opinion of researchers as to if they are two cells or still one cell. Touching cells can align themselves in a number of ways (figure 1.8).

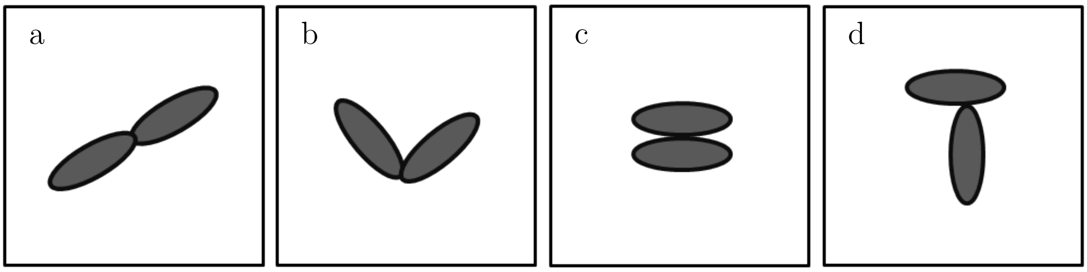


Figure 1.8: Illustration of the scenarios possible for two cells to touch.

It is important to note that removing the segmentation, and thus the analysis, of touching cells could skew the data. Cells arranged as depicted in figures 1.8c and 1.8d are never cells which are about to divide or have just divided. This means that they can be removed from analysis without skewing results providing that their alignment is not due to intercellular signals. If cells touch each other as the examples in figures 1.8a and 1.8b, their alignment could be either random in nature or due to having just divided. It is also possible that there is only one cell about to divide. Algorithms which deal with detecting cell boundaries must be reliable in correctly identifying these cases as they cannot be removed without skewing the results.

1.10 Aims and Objectives

The literature shows many algorithms designed for segmenting images from all types of microscopy images; however, it also demonstrates that the programs are not used across the biology community. This suggests that either they are not easily accessible/user friendly to encourage their use, or do not achieve the desired result for the end user. In order to produce excellent segmentation algorithms, it is important that developers have either personal experience/knowledge about the biological aspect or a large, reliable ground truth dataset. There are not many ground truth datasets available for these types of images, and the available algorithms for generating them are mainly for constructing bounding boxes.

Literature also shows that many segmentation algorithms for microscopy images also take length and width measurements, but it is not common for them to disclose the exact manner in which these measurements are made. If there is an example given, it is often an illustration of a straight, symmetrical cell. This is not ideal as cells are typically asymmetrical and not perfectly straight, so the method for measuring the cells remains uncertain.

With all this in mind, the aims and objectives of this thesis are:

- Create a manual segmentation algorithm to generate ground truth segmentations.
- Define a method to use ground truth segmentations to quantify the success of segmentation algorithms.
- Develop an automated segmentation algorithm for phase contrast/transmitted light microscopy images of *S. pombe* cells.
- Determine a more accurate and precise method for measuring the length and width of segmented *S. pombe* cells.

1.10.1 Chapter Outlines

In Ground Truth Segmentations we scrutinise quantification methods of segmentation algorithm quality and the use of 'gold standard' ground truth segmentations. We introduce a novel ground truth segmentation generation algorithm along with a method for comparing ground truth segmentations to proposed segmentations.

The second results chapter, A Novel Segmentation Algorithm for Phase Contrast/Transmitted Light Microscopy Images of *S. pombe*, we explore image segmentation algorithms and the design of a novel algorithm for the segmentation of *S. pombe*. We compare our novel segmentation algorithm to an existing segmentation algorithm with the same goal.

Chapter 5, Measuring Lengths and Widths, is where we investigate how length and width measurements are typically made and attempt to develop a novel algorithm that produces more accurate results.

Finally, the whole thesis is brought together in the discussion and conclusion chapter where we address if we were successful in achieving the aims and objectives, as well as identifying directions for further study.

CHAPTER 2

Materials and Methods

2.1 Materials

2.1.1 Chemicals

EMM	Formedium
L-glutamate	Sigma-Aldrich
Agar	Oxoid
Lectin	Sigma-Aldrich
Analytical reagent grade water	Fisher chemical

2.1.2 Yeast strains

All *S. pombe* strains used in this thesis were provided by the Mulvihill lab (University of Kent). The strains used are:

S. pombe hht-cgfp	Histone gene fused with green fluorescent protein gene.
S. pombe rlc-gfp cdc8-110	Myosin II regulatory light chain with green fluorescent protein gene and point mutation in the *cdc8* encoded tropomyosin at position 110.

2.1.3 Imaging hardware

2.1.3.1 Microscopes

IX71 inverted microscope	Olympus

2.1.3.2 Cameras

Photometrics QuantEM CCD camera	Roper
Zyla 4.2 sCMOS camera	Andor

2.1.4 Software

2.1.4.1 Image viewing and acquisition

MetaMorph [50] was used for the viewing and acquisition of the images captured specifically for this thesis.

2.1.4.2 Algorithm development

MATLAB version R2016a [68] was used for the development and testing of the proposed software algorithms presented in this thesis.

2.2 Growth and Mounting of *S. pombe* Cells for Imaging

S. pombe cells were cultured in EMMG (Edinburgh Minimal Media with glutamate as a nitrogen source) as described in [70] at 25 °C to log phase.

Cells were mounted for microscopy using lectin, as described in [48].

2.3 Viewing and Acquisition of Microscopy Images of *S. pombe* Cells

Cells were viewed on either an Olympus IX71 inverted microscope (chapter 2.1.3.1). A selection of 100x oil, 60x oil and 60x air lenses were used, with numerical aperatures of 1.45, 1.4, and 0.7 respectively. along with a variety of cameras.The cameras used were a photmetrics QuantEM CCD camera (Roper) and a Zyla 4.2 sCMOS camera (Andor) (chapter 2.1.3.2).

Images were captured using MetaMorph (Molecular Devices) [50]. The image was stored as either a 16-bit (QuantEM) or 12-bit (Zyla) image. The quantEM CCD camera images consist of a 512 x 512 pixel array with each pixel representing an area of 16 µm by 16 µm. The CMOS camera images consist of a 1024 x 1024 pixel array (for the 100x magnification) or 2048 x 2048 pixel array (for the 60x magnification) with each pixel representing 6.5 µm by 6.5 µm.

2.4 Image Datasets

All non-published datasets used in this thesis are described in table 2.1.

Dataset name	**Chip used**	**Lens used**	**Type of microscopy**	**No. of frames**	**No. of cells**	**Strain**
hybrid	CMOS & EM CCD	various	various	12	79	*hht-cgfp*
100x-oil-Q	EM CCD	100x oil	phase	119	402	*hht-cgfp*
100x-oil-Z	CMOS	100x oil	phase	200	888	*hht-cgfp*
60x-oil-Z	CMOS	60x oil	transmitted	100	391	*hht-cgfp*
60x-air-Z	CMOS	60x air	transmitted	100	203	*hht-cgfp*
fisYeast	CMOS	100x oil	transmitted	36	315	*rlc1-gfp cdc8*-110

Table 2.1: Table of datasets used in this thesis which have not been previously published.

The hybrid dataset consists of: four frames from the 100x-oil-Q dataset, two frames from the 100x-oil-Z dataset, and three frames of both the 60x-oil-Z dataset and the 60x-air-Z dataset. The fluorNuc dataset is of the GFP-labelled histones corresponding to the first 100 frames of the 100x-oil-Z dataset.

The fisYeast dataset was supplied by Irene Gyamfi (University of Kent).

2.5 Data Analysis

Chapter 3 describes the generation of ground truth segmentation for the 100x-oil- Q, 100x-oil-Z, 60x-oil-Z, and 60x-air-Z datasets. It also explains how these can be used to quantify the quality of proposed segmentations. It also describes the statistical analysis method chosen for this thesis. When ground truth segmentations are not available or usable, the proposed segments should be analysed using the criteria described in subsection 2.5.1.

2.5.1 Segmentation analysis without ground truths

Where ground truths were not available, visual determination was used to assess segmentations. To ensure as consistent an analysis as possible, segmentation assessment criteria guidelines were determined. The guidelines are described below and visual examples in figure 2.1.

- True positives - cells with a segmentation which does not deviate outside of the cell wall or the characteristic halo of this type of microscopy.
- False positives – any segmentation that does not meet the criteria of true positives.
- False negatives – any cell which does not have a true positive segmentation associated with it.

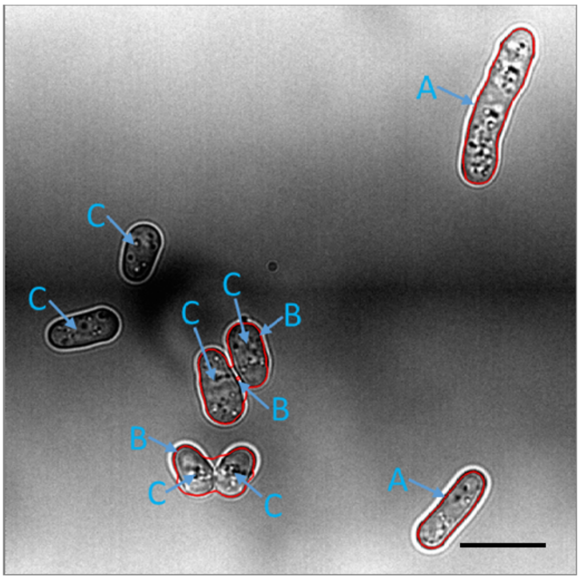


Figure 2.1: Classification of cell segmentation example. Segmentations shown in red; true positives (A), false positives (B), and false negatives (C). Scale bar represents 10 µm.

CHAPTER 3

Ground Truth Segmentations

3.1 Introduction

Development of image segmentation algorithms requires the comparison of multiple resultant segmentations of various versions of said algorithm. Once an acceptable pipeline of functions has been determined, parameters must be optimised. This results in more segmentations to compare and minute differences that may be impossible to differentiate visually. As computers are unbiased and view all images as a matrix of pixel values they are the ideal tool for identifying these differences. To do this, the computer must be informed about the desired segmentation result which has led to a number of methods for the generation of 'gold-standard' or ground truth segmentations and for the statistical analysis of these comparisons.

3.1.1 Assessing segmentation quality

Segmentation success can be measured in a number of ways, but requires a rigorous and consistent method to guarantee accurate results for comparison between segmentation algorithm iterations. For any type of segmentation algorithm, there are multiple possible techniques. The three techniques considered for this thesis are: manual visual assessment, manual ground truth segmentations, and simulated datasets.

3.1.1.1 Manual visual assessment

Proposed segmentations can be visually assessed by anyone with expertise in the field, a 'segmentor'. It is highly subjective, and with numerous iterations of the algorithm required for parameterisation this can rapidly become inaccurate, especially in borderline cases. However, once the order of functions has been set and the parameters are near to their ideal values, it is often advisable for manual visual assessment to be used to ensure that the ground truths are giving the optimal results.

There is also an issue as to the level of expertise required. If the images being segmented are of landscapes for example, it is fair to assume that the level of expertise on the subject of the image can be minimal. Issues arise when the images are of a more technical nature, such as scientific images. Unless the person who is deciding the acceptability of the images is an expert in the field in question, there is a chance that the segmentations are incorrectly rated. This can cause the final version of the segmentation algorithm to be of no use to the end user as it cannot perform the task they require.

3.1.1.2 Manually generated ground truth segmentations

Manual segmentation of images by an experienced researcher results in a set of ground truth segmentations for a given dataset. The level of knowledge required by the 'segmentor' varies depending on the nature of the images in question. Some datasets are suitable for manual segmentation by the general public via crowd-sourcing while others require very specific expertise.

Even though the computer is objective in using the ground truth dataset, manually generated ground truths are still subjective in nature. Two different experts, or even the same expert given the same dataset twice, can produce different ground truth segmentations. These segments are likely to be very similar, especially when visually compared, however, a computer will give different results depending on the ground truths provided.

3.1.1.3 Published ground truth datasets

There are several readily available datasets with ground truth segmentations supplied such as the Berkeley Segmentation Dataset and Benchmark [71] and the MIT Intrinsic Images datsaset [72]. The problem with these datasets is in the type of images supplied – most are everyday photographs/images and of the biological datasets available, few are of microscopy images and only one is of fission yeast cells.

The PombeX dataset [73] is of fission yeast bright field microscopy images; however, there are only eight unique ground truth images with a total of 1,422 cells. As discussed in chapter 1, cells at densities above log phase are no longer in optimal growing conditions which will affect their growth and therefore are often inappropriate for biological studies. Growth and mounting conditions are not provided by either the dataset or in the paper from which the images are taken [62, 73]. This means that any algorithm developed from using them may not perform well on images that are grown or mounted using different conditions and these conditions cannot be replicated to assist the algorithm.

3.1.1.4 Generating ground truth segmentations

Plenty of programs exist online to produce your own ground truths [74, 75, 76]. The processes used by these programs often include bounding boxes [74, 75] or allow users to create polygonal regions [74]. This makes them unsuitable for segmenting cells that are predominantly curved in shape. Many of these are acceptable for their predominant use as the segments they are going to be compared to do not need to be precise as measurements are not going to be taken from them. An example of this is of road signs. The segmentation algorithm must identify them, however when comparing the results, a bounding box will suffice in checking whether a suitable area in the relevant region has been segmented. Usually, biological data requires measurement to be taken and therefore precision is crucial.

Few programs produce truly smooth edged segments. One such program that was designed specifically for biological image segmentation is ITK-SNAP [76]. The detailed boundaries given by ITK-SNAP would be ideal for manually segmenting cells, however, it is too specialised. Two-dimensional images cannot be used with the program and it is extremely unusual to take a stack of phase microscopy images.

Since there are no readily available manual segmentation algorithms suited for images of cells, a new method to generate ground truths is required. Ideally, this new method will segment straight and curved lines ending in curved or pointed edges. These features can be used to construct all possible shapes. If these characteristics can be formed, all shapes that may ever be encountered can be replicated and the algorithm will be genuinely multipurpose.

Basic/basis spline (B-spline) curves are perfect for this and have even been used for part of biological image segmentation [77]. The creation of these curves begins with ‘knots’ or co-ordinates being selected. From these knots polynomials are fitted and then subjected to a smoothness function. The smoothing function works on the all of the polynomials to ensure a contiguous shape with effortlessly merged joints. This results in a closed segment with seamless transition between all knots.

There is always a unique solution with minimum energy, guaranteeing the exact same output shape for a set of defined knots. This means the appropriate selection of knots is vital. If the chosen knots are unsuitable, the desired output shape will not be replicated. To create sharp corners, selecting a knot twice will result in a point; straight lines can be produced by selecting several knots in a line.

3.1.1.5 Simulated data ground truths

Computer generated simulated data is another option for creating ground truth images with a matching dataset. Programs are freely available online to produce any number of shapes with various features, from objects in a box [78] to microscopy images [79, 80, 81]. The computer creates a ground truth segmentation mask from the input parameters provided by the end user and then artificially alters them to emulate images acquired naturally.

Datasets of cell microscopy image can be developed using various modelling methods [79, 80, 81]. The first step is to define the cell shape, either by entering parameters [79, 80, 81] or by a machine learning algorithm [79, 80]. The second step is to replicate the texture of microscopy images. For fluorescent images this is easier than it is for phase contrast images because of ‘optical aberrations generated by the imaging system’ preventing the true texture of the cell from being observed [79, 80].

Next, the density and probability of overlapping cells has to be set. This is done by the user selecting parameters. Varying the parameters between datasets allows for rigorous testing of segmentation algorithm success in different conditions. These images are then degraded to simulate real images, mimicking the effects of sample preparation and imaging conditions [79, 80].

Imitating realistic cell shapes is easy enough for such programs; however, emulating the background noise and artefacts characteristic of phase contrast microscopy is harder. One program for generating simulated datasets, SimuCell [81], claims to have three benefits over previous programs: it makes diverse heterogeneous populations; it can adapt to interdependencies 'among population, biomarker and cell phenotype'; and, it has a 'palette' of phenotypes which users can easily add to.

3.1.2 Quantifying segmentation quality

Where ground truth segmentations exist, and can be compared to segmentation algorithm outputs, a measure needs to be made to quantify the quality of the segmentations. Quantification means that various iterations of an algorithm can be compared numerically, as well as allowing comparison of different algorithms to each other. There are two features that can be used to compare segmentation quality: segment perimeters or segment areas. There are multiple quantification techniques, with each performing differently dependent on the feature.

3.1.2.1 The Jaccard index

The Jaccard index is a simple and effective method to compare ground truth (G) and proposed (P) segments. There are two potential ways to implement the Jaccard index (equation 3.1): either considering the segmentation masks as single images, or comparing all of the segments individually.

$$J = \frac{|P \cap G|}{|P \cup G|} \tag{3.1}$$

Complete, individual segments are typically considered [82, 83, 84], but it is not unheard of for perimeter pixels to be compared [83]. Both over- and under- segmentation are equally penalised in this method. To ensure segments are compared accurately, the ground truth and proposed segmentation masks must isolate segments of interest. This can lead to problems when there are multiple segments in one image that correspond to one segment in the other.

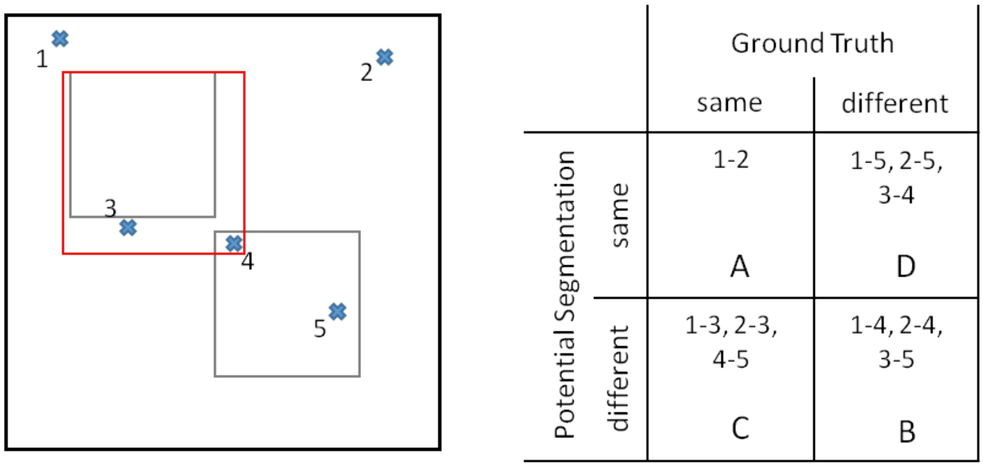
Unless the segmentation masks are considered as a single image, the Jaccard values must be further manipulated to give one value for each image. One option is to average the best Jaccard value for all of the ground truth segments [85], however, this means that a missed cell could potentially have the same effect as several poorly segmented cells. This is not ideal as it does not discern between inaccurately segmented regions and missed segments.

3.1.2.2 The Rand index

Another common method is the Rand index [82, 83, 86, 87]. To calculate the Rand index (R) [88], all pixels are taken in pairs and compared between the ground truth segmentation mask and the proposed segmentation mask. A tally is made from the results of pairs of pixels where: A is when the two selected pixels are in the same segment in both masks; B is when the two selected pixels are in different segments in both masks; C is when the two selected pixels are in the same segment in the ground truth segmentation mask, but different segments in the proposed segmentation mask; D is when the two selected pixels are in different segment in the ground truth mask, but the same segment in the proposed mask (figure 3.1). The number of pixels in each group is then used in equation 3.2.

$$R = \frac{A + B}{A + B + C + D} \tag{3.2}$$

If ground truth segmentations were manually generated, the resultant masks will be subjective in nature. A leniency may be desired for the edges of segments to allow visibly acceptable proposed segmentations to be rewarded. The normalised probabilistic Rand (NPR) index was developed to overcome this shortcoming [87]. It uses the principle idea from the Rand index and imposes a probability distribution over the ground truth segments to weight pixels,



		Ground Truth	
		same	different
Potential Segmentation	same	1-2 A	1-5, 2-5, 3-4 D
	different	1-3, 2-3, 4-5 C	1-4, 2-4, 3-5 B

Figure 3.1: (L) Diagram of ground truth image (grey) and potential segmentation (red) with five 'pixels' marked (blue). (R) Table showing pixel pairing classification in regards to the Rand index.

penalising blatant misses more severely than borderline misses. The resultant values are then normalised by equation 3.3.

$$\text{Normalised index} = \frac{\text{Index - Expected index}}{\text{Maximum index - Expected index}} \tag{3.3}$$

3.1.2.3 Consistency errors

The global consistency error (GCE) and local consistency error (LCE) are often used for rating segmentation algorithms [86, 87, 89, 90]. For both, the local refinement error (LRE) is calculated for all segments by equation 3.4 (except in [89] where they multiply the LRE by the intersect of $S_i(P)$ and $S_j(G)$ for the Martin Error Measure (MEM) which is substituted into the GCE and LCE in place of the LRE), comparing the ground truth segmentations (G) to the proposed segmentations (P) and vice versa. The GCE then sums the LREs for each direction of comparison and selects the minimum value (equation 3.5), while the LCE selects the smallest LRE for each segment and sums the values together (equation 3.6).

$$LRE(S_j(G), S_i(P)) = \frac{|S_j(G) - S_i(P) \cap S_j(G)|}{|S_j(G)|} \tag{3.4}$$

$$GCE(G,P) = \frac{1}{N} min \left\{ \sum_{j=1}^{X} \sum_{i=1}^{Y} LRE(S_j(G), S_i(P)), \sum_{j=1}^{X} \sum_{i=1}^{Y} LRE(S_i(P), S_j(G)) \right\} \tag{3.5}$$

$$LCE(G,P) = \frac{1}{N} \sum_{j=1}^{X} \sum_{i=1}^{Y} min \big\{ LRE(S_j(G), S_i(P)), LRE(S_i(P), S_j(G)) \big\} \tag{3.6}$$

where X is the number of segments in G, Y is the number of segments in P, with j and i as counters for the segments of X and Y respectively. $S_n(I)$ denotes the nth region of the segmentation I. The total number of pixels in the image is given by N.

Evaluation by LCE and GCE does not penalise over-/under-segmentation. This means the more pixels that are common to both segmentations, the better the score presented, regardless of the actual similarity between the segmentations. In an attempt to improve on this the object-level consistency error (OCE) was developed [89].

The OCE attempts to penalise over-/under-segmentation. It uses the partial error measure (PEM; equation 3.7) and selects the minimum value from PEM(G,P) and PEM(P,G). The W_{ji} and W_j coefficients ensure only segments that interact with the relevant segments are considered (W_{ji}, equation 3.8) and weights the score based on the proportion of pixels in the segment and the total number of segmented pixels (W_j, equation 3.9) [89]. This means that over- and under- segmentation are penalised, however, the penalty is heavily dependent on the size of the segment compared to the total area of segments. One large segment that has been poorly segmented can have a large impact on the OCE.

$$PEM(G,P) = \sum_{j=1}^{X} \left(1 - \sum_{i=1}^{Y} \frac{|S_j(G) \cap S_i(P)|}{S_j(G) \cup S_i(P)|} \cdot W_{ji} \right) \cdot W_j \tag{3.7}$$

$$W_{ji} = \frac{\overline{\delta}\big(S_j(G) \cap S_i(P)|\big) \cdot |S_i(P)|}{\sum_{k=1}^{X} \overline{\delta}\big(|S_j(G) \cap S_k(P)|\big) \cdot |S_k(P)|} \tag{3.8}$$

$$W_j = \frac{|S_j(G)|}{\sum_{j=1}^{X} |S_j(G)|} \tag{3.9}$$

where j/k and i are counters for segments of X and Y respectively. The $\overline{\delta}$ (x) gives a value of 0 when x is equal to 0, and gives a value of 1 otherwise.

3.1.2.4 Precision, recall and F score

Alternatively, an F score (F, equation 3.12) can be used [86, 91, 92]. It relies on both precision (p; equation 3.10) and recall (r; equation 3.11) to be calculated first. Sometimes, an F score is not calculated but the precision and recall values are used [83, 84]. However, this does not result in one number for direct comparison, which can lead to ambiguity in the results.

$$p = \frac{TP}{TP + FP} \tag{3.10}$$

$$r = \frac{TP}{TP + FN} \tag{3.11}$$

$$F = \frac{2 \cdot p \cdot r}{p + r} \tag{3.12}$$

Precision and recall require totals for the number of true positives, false positives and false negatives. This can be considered on a pixel level or a segment level using any number of methods, including the Rand index or Jaccard index described above.

3.1.2.5 Comparing methods by segmentation feature

Comparing the location of edge pixels for the proposed and ground truth segmentations is done in two ways. The first way is to find the ratio of common boundary pixels and missed boundary pixels. This can be done using the Jaccard index (equation 3.1) [83], or variants of it which determine precision and recall [82, 86] . The second way is to measure the distance between the proposed segmentation's perimeter and the ground truth's perimeter using a nearest neighbour algorithm. The distance between pixels can then be used to make a histogram [93] which can then be compared for different segmentations.

One of the biggest problems faced by boundary matching methods is that the width of the boundary affects the results. It also affects the likelihood that the exact same pixels are selected. Using any of the Jaccard index-based methods can result in misleading values. Visually acceptable segments, such as those seen in figure 3.2a, will result in a score of zero whilst segments that would be rejected upon visual assessment (figures 3.2b and 3.2c) provide non-zero scores. This is why [83] suggests 'blurring' the perimeter pixels before comparison. While comparing histograms of pixel distances overcomes this, it is difficult to determine the meaning of histogram features.

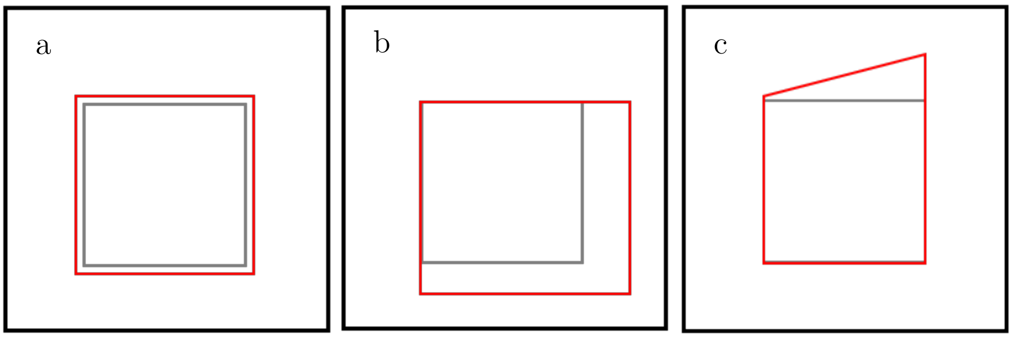


Figure 3.2: Diagram of ground truth image (grey) and potential segmentation (red). Perimeter Jaccard index scores of (a) 0.00, (b) 0.29, and (c) 0.67.

Considering segmentation areas is often deemed a more thorough and rigorous metric as the results are no longer reliant on perimeter width. All of the methods are theoretically sound; however, depending on the specific problem at hand, some methods are better suited than others.

Jaccard's index is also used for comparing segmentation areas. It is a more resilient measure when used on the whole segment's pixels as opposed to only the perimeter pixels as it is not dependent on the perimeter's width. The consistent penalisation of over-/under-segmentation makes this an ideal method for determining the number of true positive, false positive and false negative segments if an 'acceptance' threshold is established, however, the values themselves are not helpful in rating segmentation algorithms.

The Rand index and its variants are computationally more expensive than the Jaccard index. Because of this it is rarely used to count the number of true positive, false positive and false negative segments. On its own, the Rand index is more informative than the Jaccard index as to the segmentation success of an algorithm; however, as background pixels influence the score, the Rand index is affected by segment density. The NPR is potentially more interesting due to the use of a probability distribution to weight each pixel's contribution to the overall score, but there is no universally accepted distribution.

Consistency errors are commonly used, but unless the primary segmentation criteria is to ensure all pixels of the ground truth segmentation are segmented in the proposed segmentation the GCE and LCE are not beneficial. The OCE penalises for over-/under-segmentation, however, it is best suited to segmentation problems where all segments of a proposed segmentation are considered individual targets (i.e. no background pixels).

An F score combines the precision and recall properties of a segmentation algorithm. It can be an ideal method for combining the Jaccard index results into one comparable value as it can be weighted depending on the importance of precision and recall. The ability to separate the importance of precision and recall make this technique adaptable to the present situation.

3.1.3 Aims and objectives

The aims and objectives of this chapter are:

- Develop an algorithm for highly detailed ground truth segmentations.
- Generate ground truths for five datasets of fission yeast phase microscopy images.
- Determine a suitable method for using the ground truths to help develop a novel image segmentation algorithm by assessing segmentation quality.

3.2 Methodology

3.2.1 Previously described datasets

This chapter uses the 100x-oil-Q, 100x-oil-Z, 60x-oil-Z, 60x-air-Z, and hybrid datasets described in chapter 2.1.

3.2.2 Algorithms for generating ground truth segments

To create the ground truth segmentations, two algorithms were written. The first algorithm (GTGen) allows an expert to manually segment the ground truths. Due to the tedious and repetitive nature of the task, a second algorithm (GTVer) was considered to be beneficial. This second algorithm allows the segmentor to verify their segmentations and redo any that they wish to.

Both algorithms were designed and run in MATLAB. The current directory should be set to the folder containing the images the user desires to make ground truths for. Once the programs run, the outputs are saved in the directory as a

MATLAB structure. The ground truth structures, along with the input images and the necessary lines of code to convert the structure into segmentations, are available at [94].

3.2.2.1 GTGen

For the initial manual segmentation GTGen was developed. Figure 3.3 shows the sequence of steps undertaken by the algorithm that, with user input, create the initial ground truth segmentations.

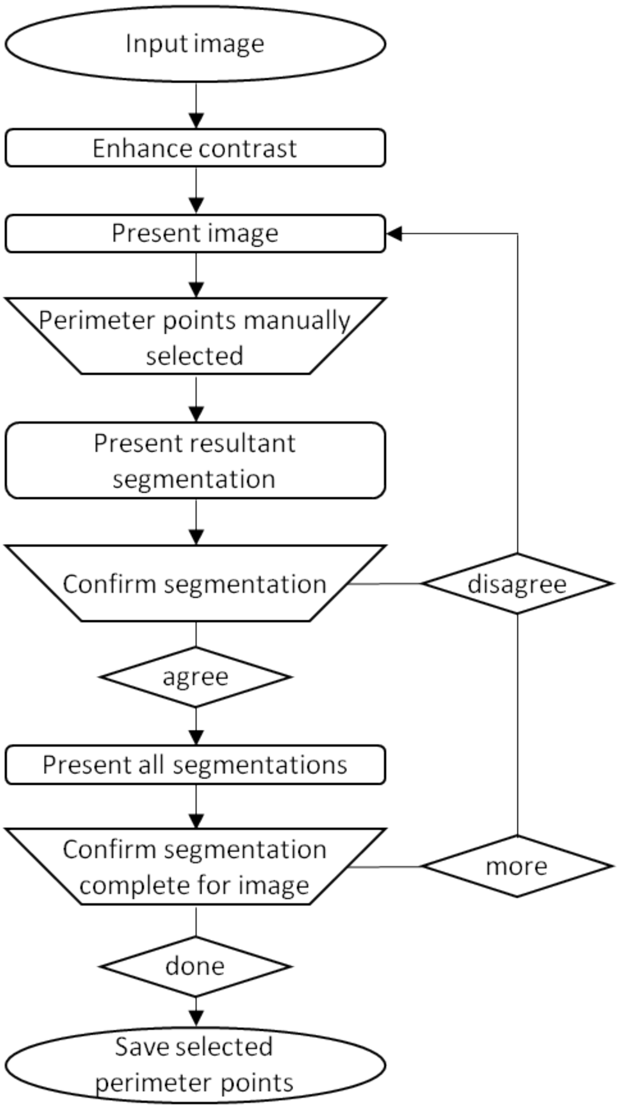

Input image
Enhance contrast
Present image
Perimeter points manually selected
Present resultant segmentation
Confirm segmentation
disagree
agree
Present all segmentations
Confirm segmentation complete for image
more
done
Save selected perimeter points

Figure 3.3: Schematic of GTGen algorithm for the production of ground truth segmentations.

Each image of the dataset is taken in turn, the contrast is maximised and the resultant image displayed. The segmentor is then permitted to manually select points for a single region of interest (in this thesis, a cell). Because B-splines are constructed from ordered co-ordinates, the sequence that the points are selected is important. The user should select points in either a clockwise or anticlockwise direction until the initial loop has been completed. After this has been accomplished, extra points can be added if required. Repositioning of the knots is possible until the user saves the selection by double clicking. The points are then saved in the output structure (called GroundTruthStructure).

3.2.2.2 GTVer

As manual segmentation is subjective and time consuming, the quality of the segmentations can fluctuate throughout a dataset. GTVer allows the user to verify their segmentations to ensure consistent segmentations over the whole dataset and redo poor segmentations caused by fatigue (figure 3.4).

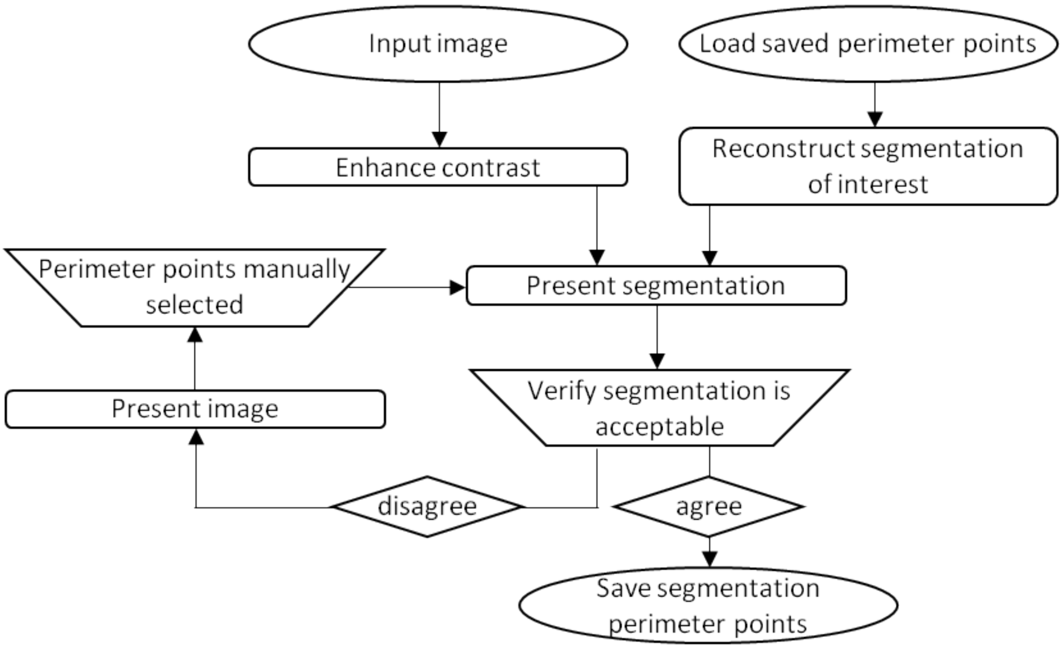

Input image
Load saved perimeter points
Enhance contrast
Reconstruct segmentation of interest
Perimeter points manually selected
Present segmentation
Present image
Verify segmentation is acceptable
disagree
agree
Save segmentation perimeter points

Figure 3.4: Schematic of GTVer algorithm for the verification of ground truth segmentations generated by GTGen.

The co-ordinates stored in the structure, GroundTruthStructure, along with the original images are used to display the segmentations one by one. When a segment is shown, a prompt in the command window asks if the segmentation is to be accepted. If a new segmentation is desired, the contrast adjusted image is displayed and the segmentor can select new co-ordinates in the manner as in GTGen. The current segmentation is displayed for verification and once accepted, the algorithm moves on to the next segmentation until all have been completed. It is important for the user to be vigilant and only select new knots for the cell that they are currently verifying.

3.2.3 Selecting a quantifying method

As shown in section 3.1.2, there are numerous methods that can be used to quantify the quality of proposed segments. An F score was chosen to produce a single value for each algorithm and its iterations which were then directly comparable. To identify true positive, false positive and false negative segmentations of cells, the Jaccard index value (J, given by equation 3.1) can be compared to a range of pre-determined thresholds. The range for the 'true positive' threshold was selected after the analysis of several sets of manual segmentations for a subset of images.

3.2.3.1 True positive threshold range

To determine the range of values to use as the threshold for true positives, a simple experiment was performed:

- Five individuals with biosciences experience manually segmented each cell in the hybrid dataset.

- The segmentations were saved for each individual and for each segmentation, a composite segmentation was made by taking the average response from the set of segmentations.
- The composite segmentations were compared to the ground truth segmentations, using GTTestMod1 (as will be described in section 3.2.4.2).
- The Jaccard index was calculated and recorded for each segment, with the mean and median computed for each individual and for each segment.

The results of this experiment are shown and evaluated in section 3.3.

3.2.3.2 Weighting of the F score

Two things are needed to calculate an F score: a value of precision (equation 3.10) and a value of recall (equation 3.11). It can also be weighted to give precedence to either precision or recall. This preference is taken into account by selecting a value for β with it being the importance of recall over precision – i.e. $\beta = 2$ means recall is twice as important as precision while $\beta = 0.5$ means precision is twice as important as recall (equation 3.13).

$$F_\beta = \frac{\left(1 + \beta^2\right) \cdot p \cdot r}{\left(\beta^2 \cdot p\right) + r} \tag{3.13}$$

3.2.4 Evaluating proposed segmentations with ground truth segmentations

To use the ground truth segmentations to assess the quality of segmentations proposed by an algorithm, the ground truth segmentations need to be reconstructed. The reconstructed segmentations are then compared to proposed segmentations by any method the user desires.

3.2.4.1 GTTest

To compare the ground truth segmentations to the proposed segmentations, another algorithm is needed. This algorithm has to be easily adaptable to allow the end user to modify it so that it best suits the format of their output segmentations. This is problematic for comparing existing segmentation algorithms to novel algorithms as the source code is rarely readily available to modify.

For use with segmentations where the segmentation algorithm code is fully accessible, GTTest is suitable. The ideal segmentation output for GTTest is either individual masks for each segmentation or a mask where the segmentations which do not touch. This ensures that the correct proposed segments are compared to the relevant ground truth segment.

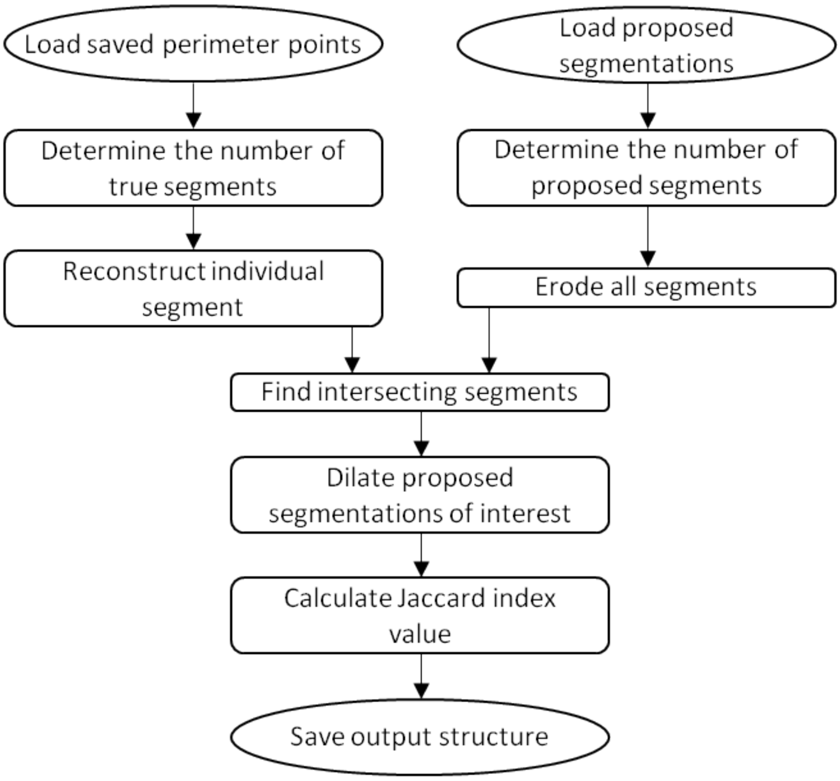

Load saved perimeter points
Load proposed segmentations
Determine the number of true segments
Determine the number of proposed segments
Reconstruct individual segment
Erode all segments
Find intersecting segments
Dilate proposed segmentations of interest
Calculate Jaccard index value
Save output structure

Figure 3.5: Schematic of GTTest algorithm for the comparison of ground truth segmentations generated from GTGen to proposed segmentations from the output segmentation mask.

The basic code, as shown in figure 3.5, loads the ground truth structure, reconstructs an individual segmentation's perimeter and converts it into a mask. This mask is then compared to the eroded proposed segmentations which have been labelled by MATLAB's inbuilt function, bwlabel. Any touching segments from the proposed mask are then treated individually. The proposed segment is dilated with the same instructions as the previous erosion, and the Jaccard index is calculated and stored.

Users can modify the source code to allow their individual proposed segmentations to be considered. One suggested method would be to import the binary mask of all the segmentation outlines, fill the segments, and subtract the outlines from the mask. The potentially matched segments would then need to be dilated again before comparing to the ground truth segment to account for the loss caused by the subtraction of the outline.

3.2.4.2 Modifications of GTTest

In this thesis, a couple of modified versions of GTTest are used. These modifications were required for: comparing manual segmentations to other manual segmentations (GTTestMod1); and comparing proposed segmentations to the ground truth dataset provided by PombeX [73] (GTTestMod2).

For GTTestMod1, there are two structures of co-ordinates. This means that the proposed segmentation is not from the final output image and the first three of steps of the right column of figure 3.5 are replaced with those from the left column. As the sets of knots for each segmentation have been manually ordered to replicate the order of the ground truth segmentations for each image; the comparison algorithm can assume the presented segmentation masks are of the same cell, and therefore, can be compared.

GTTestMod2 was used with the PombeX ground truth dataset [73]. In this dataset, the ground truths are saved as mask images, with one ground truth image per original image. Here, the ground truth input renders the first three steps on the left of figure 3.5 irrelevant. The provided ground truth mask can be labelled by bwlabel and each segment from the ground truth image is taken in turn and follows the rest of the steps of GTTest as shown in figure 3.5.

3.3 Results and Discussion

3.3.1 Using ground truths

Ground truths are the favoured method for comparing segmentation algorithms because they allow for objective and consistent comparisons. Before a computer can analyse proposed segmentations, the ground truths must first be established either by manual segmentation or simulated data generation.

3.3.1.1 Ground truths from simulated data

Many simulated data generation programs are freely available online [81, 95], meaning theoretically there is an unlimited supply of images. There is a common problem faced by all of these programs, namely that it is difficult to successfully recreate artefacts from imaging systems [81, 96, 97]. This is more of a problem for phase microscopy images than it is for fluorescent microscopy images due to the characteristic typically associated with their backgrounds. Both SimuCell and OpenMx are complicated algorithms and lack comprehensive instruction manuals.

3.3.2 Novel ground truth generation algorithm

As polygonal shapes cannot yield curved sections, a different method is needed to allow for accurate manual segmentation. While the time commitment required for manual segmentation can never be eliminated, it can be reduced. These were the two areas which GTGen and GTVer were intended to improve.

In the designing of GTGen and GTVer, a method was required to recreate all possible shapes which may be encountered by a segmentation algorithm. B-splines create polynomial sections and are therefore ideal. Zeroth order polynomials give a single point; first order polynomials give straight lines; while second order and higher polynomials produce a variety of curves. This means that both straight and curved edges (including curved corners) are fitted with non-zeroth order polynomials and sharp points are constructed by zeroth degree polynomials. With these features, all possible shapes can be mapped.

While fitting straight lines now requires the selection of additional co-ordinates, the extra knot is less time consuming to select than the many co-ordinates previously required to replicate curves. The number of knots needed to mimic an arc depends on the curvature of the curve. With practice, this can be estimated adequately, reducing the number of re-segmentations required. This reduction in user input significantly minimises the length of time required for the segmentation of each dataset.

Knowing that segmentations can be easily returned to and corrected if they are unsuitable saves time, this is the main benefit of GTVer. Often a segmentor will assume that they have been consistent over an entire dataset, however, when reassessing their segmentations, they often find that they were not. This is especially true for larger datasets which may take several days to complete. GEDI and other segmentation programs do not have this feature specifically, however, as with GTGen, it is possible to re-segment a cell. Having a simple method for verifying segmentations is particularly helpful, even if just to confirm the validity of the segmentations.

3.3.3 Datasets with ground truths

Using GTGen and GTVer, the ground truths for four different datasets of phase microscopy images of fission yeast cells were produced. Each of these datasets has different magnifications, immersion mediums, and cameras while encompassing 1883 cells over 519 images. The images and segmentations are freely available from Kent Data Repository [94] and can be used by anyone in any manner they deem appropriate. This means that they are able to select their preferred method for comparing segmentation algorithms or use the method described in this thesis.

These datasets are more versatile and comprehensive than the PombeX ground truths dataset. The PombeX dataset has densely populated images which are not representative of the images often taken for biological studies, and the imaging conditions are not specified. This, along with the fact that there are only eight images, makes it a limited dataset.

3.3.4 Quantifying quality assessment

Selecting a method for quantifying the quality of segmentations is important, and the problem at hand dictates the most favourable solution. The main purpose for these segmentations is to measure the geometric properties of each cell; the lengths and widths of cells, along with other such properties. This means that a high degree of segmentation accuracy is desirable; however, if segmentations, and therefore measurements, are not precise the resulting averages of a mutant's phenotype could be drastically incorrect.

3.3.4.1 Choosing a quality quantification method

Cells are discrete objects within the frame and the background is not of interest, so it can be considered that there are no true negative pixels. Many methods also treat the frame as a whole and compare segmentations on a pixel basis. This means that the area covered by cells can affect the resultant score of the algorithm. Because of this, it was decided to treat the segments as individual objects; categorise them as either true positive, false positive or false negative; and then use the number of cells in each category to rate the segmentation algorithm.

An F score is well suited for rating segmentation algorithms on a segment level basis, especially as it can be weighted in favour of either precision or recall. For cell segmentation algorithms, it is important for cells to be correctly segmented. A missed cell is less problematic than an inaccurate segmentation as it requires less time and effort to acquire more images but inaccurate segmentations (or false positives) demand the time of the researcher to remove it as well as potentially requiring further imaging. As precision is more important than recall for automated cell segmentation algorithms, an $F_{0.5}$ score was chosen ($\beta = 0.5$ in equation 3.13), which makes precision twice as important as recall.

3.3.4.2 Categorising segments

Many of the methods for evaluating segmentation algorithms can all be used to categorise segments as true positives, false positives or false negatives. Once the number of true positives has been determined, the number of false positives is the total number of proposed segments minus the number of true positives, and the number of false negatives is the total number of ground truth segments minus the number of true positives.

Even though cell segmentation often involves edge detection algorithms, quantifying how similar proposed segmentation's perimeters are to the ground truth's perimeter is not ideal. This is because of the exacting nature of their analysis (discussed in section 3.1.2). Having dismissed perimeter matching methods based on the problems highlighted in figure 3.2, there are several area matching methods still to consider.

The Rand index is computationally extremely expensive. To make this comparison on a segment-by- segment level, either the segments would have to be cropped or the analysis would take a long time. Cropping the image poses a problem as the proportion of background pixels to object of interest affects the rating. The segments would also have to be matched between the ground truth mask and the proposed segmentation mask.

The GCE and LCE are both reliant on the comparison of all segments to one another. This is not helpful when individual segment scores are desired. The OCE only considers segments that overlap, but it considers the entire frame. Using the PEM instead means that individual scores could theoretically, be used to categorise segments. However, it would require modification by removing the summation of all segments for a frame. This makes it a variation of the LRE and of Jaccard's index. It still requires a significant amount of processing power to isolate each segment, compare it to all the other segments in the complimentary segmentation mask, and calculate the PEM, making it unsuitable.

A much simpler method is using the Jaccard index (equation 3.1). Only the ground truth segments need to be isolated to create individual masks. These masks can then be used to identify segments that overlap in the proposed segmentation mask and the Jaccard value calculated. This is computationally inexpensive, making the evaluation process faster.

3.3.5 Human participation study design, results and discussion

To use the results from the Jaccard index calculations, a threshold has to be set. To maximise the usability of this method, a suitable range of results would be preferential. This is because segments produced during the development of a segmentation algorithm may be more promising for the desired end goal but require parameter optimisation to better 'fit' the cell. As with the 'expected value' used by the NPR index (equation 3.3), arbitrarily selecting these values is not ideal. To overcome any potential prejudices, an experiment was devised to determine a suitable threshold level (described in section 3.2.3.1).

	J value	
considered segments	**mean**	**median**
100x-oil-Q	0.88	0.89
100x-oil-Z	0.90	0.91
60x-oil-Z	0.69	0.69
60x-air-Z	0.76	0.78
60x mag	0.72	0.72
100x mag	0.89	0.90
EM CCD camera	0.88	0.89
CMOS camera	0.82	0.86
air immersion medium	0.76	0.78
oil immersion medium	0.85	0.89
all segments	0.84	0.88

Table 3.1: Table of mean and median Jaccard index values from the human participation study using the hybrid dataset. Scores provided for: each contributing dataset, each variable imaging condition (magnification, camera, and immersion media), and the complete hybrid dataset (all segments).

3.3.5.1 Results of the human participation study

The means and medians for the comparison between ground truth and composite segmentations of each imaging condition are shown in table 3.1.

3.3.5.2 Experiment design

The study had five participants which is a small sample size; however, all of the segmentors have the relevant knowledge. Increasing the sample size with participants with less extensive knowledge would be damaging to the results, as they are not qualified to make such a judgement.

The means and medians of the Jaccard index values produced from comparing the composite and ground truth segmentations show several things. The first of which is that the type of camera did not affect the scores. The 100x-oil-Z and 100x-oil-Q showed no significant differences (table 3.1).

Secondly, the Jaccard values for the 100x magnification images are significantly better than the 60x magnification images ($\geq$0.89 and 0.72 respectively, table 3.1). This could be simply explained by a higher magnification increasing the size of the segments and the larger regions are easier to segment with a high level of certainty. The larger regions can also skew the results as the intersecting area is likely to be larger proportionally to the total area (union of the segments).

Furthermore, the difference between the images with oil or air as an immersion media is minimal compared to the magnification. The values for the mean and median for the air and oil segmentations span a range of 0.13 (table 3.1). Fatigue could be partially to blame for this as the participants segmented 54 cells from the 100x magnification images before the air images were segmented. There are also no 100x magnification air images, and as the 100x magnification had a much better score than 60x magnification, this could also influence the outcome.

It is also important to note that some of the participants segmented touching cells as a singular cell. This would skew the means but the medians would remain stable. The incorrect segmentations were predominantly results of uncertainty or a difference of opinion; however, some were caused due to a break in focus. This highlights the difficulties of manual segmentation and the need for a verification algorithm such as GTVer.

3.3.5.3 Determining the range of the true positive threshold

Ideally, the threshold for determining true positives should be in line with the level of disagreement between segmentors. This means that if manual visual assessment accepts a segment as a true positive and it is calculated that on average 75% of the ground truths' pixels are included in the proposed segments of accepted segmentations, a computer should use 75% as the threshold. This means two things must be considered: should the mean or median be used to calculate the average; and will one threshold value for all datasets suffice.

As the Jaccard values for the 60x magnification images are significantly different to that of the 100x magnification images, visual assessment was carried out on the individual segmentations. This showed that the participants were less accurate in their segmenting, possibly due to fatigue, but perhaps also owing to the level of detail from the magnification. Because the visual assessment agreed that the Jaccard values were reliably different for the 60x magnification image compared to the 100x magnification images, it was decided that one threshold value would suffice.

The mean and median of the Jaccard values from the comparisons between all of the composite and ground truth segments are different (mean = 0.84, median = 0.88; table 3.1). The mean averages of all results include those of the touching cells which were not always segmented as individual cells. This can skew the results and makes it a less reliable measure of how similar manual segmentations are, so the median would be a better measure for the threshold as it removes any outlying segmentations. Therefore, the threshold values should be based around 0.88 (table 3.1) and a range of 0.75 to 0.95 in increments of 0.05 was chosen to identify true positives.

3.4 Conclusions

In the chapter, I describe:

- Development of an algorithm for highly detailed ground truth segmentations.
- Generation of ground truths for five datasets of fission yeast phase microscopy images.

- Determination a suitable method for using the ground truths to help develop a novel image segmentation algorithm by assessing segmentation quality.

The development of GTGen and GTVer means that there is a multi-purpose algorithm for detailed manual segmentation. Both algorithms were used to create ground truths for the five datasets of fission yeast phase microscopy images. A suitable method has also been established to use the ground truth segmentations in the development of a novel segmentation algorithm for this thesis.

CHAPTER 4

A Novel Segmentation Algorithm for Phase Contrast and Transmitted Light Microscopy Images of *Schizosaccharomyces pombe*

4.1 Introduction

4.1.1 Image segmentation algorithm pipeline design

In chapter 1.8, the common techniques of image analysis algorithms were introduced. The first key point is that neural networks and machine learning algorithms are sometimes used in segmentation algorithms; however, they are not ideal for phase contrast/transmitted light microscopy image segmentation algorithms. Furthermore, traditional image segmentation algorithms are constructed of functions which can be organised into four main groups: image enhancement, morphological filtering, image segmentation, and region analysis [67].

4.1.1.1 Image enhancement

Image enhancement is primarily concerned with removing and/or reducing background noise. It can be thought of in two main methods: contrast enhancement, and filtering.

Contrast adjustment

There are three main methods to adjust the contrast of an image: histogram equalisation, stretching the range of the pixel values, and converting the greyscale image into a binary image.

Histogram equalisation works by calculating the histogram of the pixel values and then altering the pixel values so that they match the value assigned to the bin they fall into. In MATLAB, this can be achieved using the histeq command. The size/number of bins used for the histogram determines the level of detail present in the output image, but there is always a loss of detail when using this method.

Stretching the range of the pixel values to cover the entire possible range of pixel values means that the number of unique values stays the same, however, the contrast between pixels of different values is maximised (figure 4.9). The inbuilt function, imadjust, in MATLAB is suitable for this. This type of stretching ensures the most visually clear result and ensures no loss of data.

Converting a greyscale image to a binary image is a severe method for adjusting the contrast. The im2bw command in MATLAB does this by determining if a pixel becomes black or white based on the value set as its threshold. This method is the most extreme in the sense that the pixels suddenly only have an option of two values and therefore fine detail is automatically lost (figures 4.1c and 4.1d). The threshold used by im2bw can be arbitrarily set (figure 4.1c) or can be determined via Otsu's method (using the function graythresh; figure 4.1d). Otsu's method is designed to automatically select the optimal threshold for im2bw using only the grey-level histogram of the image [98].

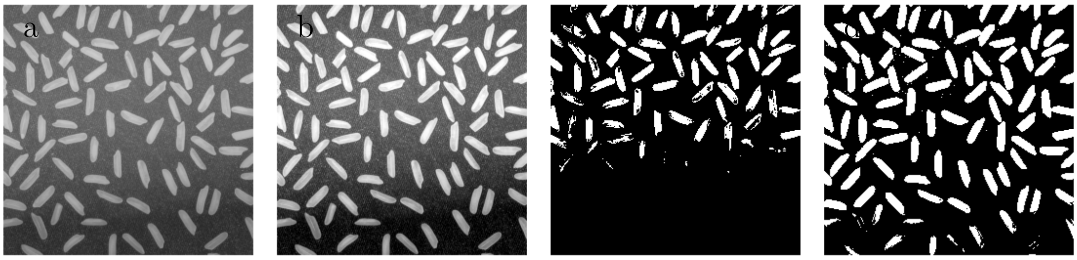


Figure 4.1: Image contrast enhancement examples: (a) input image; (b) maximally stretching the histogram; (c) arbitrarily setting the threshold to 0.7; (d) threshold set using Otsu's method (determined to be 0.5137).

Filtering

Filtering reduces noise by changing the value of pixels based on the values of neighbouring pixels. Filter kernels define the process through which the pixel values are changed. The simplest kernels average the neighbourhood using either the mean or median. More advanced kernels weight pixels unevenly within the neighbourhood depending on their position. This results in specialised filters such as Gaussian [99] or edge-emphasising filters like Canny, Prewitt, Sobel, or Hessian [100, 101, 102, 103]. Due to the averaging of values, the image becomes slightly blurred and the aim is to blur the background noise to make it more uniform in value without losing too much detail of the object(s) of interest.

Adaptive filtering is where the filter determines if the variance of the neighbourhood is large, and if it is, the smoothing is minimised to maintain the level of contrast [104]. The goal is to blur the background noise by maximising the averaging around pixels which are not close to edges, as defined by contrast borders, while maintaining the sharp contrast at the border.

4.1.1.2 Morphological filtering

Morphological filters also aim to remove and/or reduce background noise. There are many functions that can be used such as imdilate, imerode, and bwareaopen among others. Some of these filters use 'structural elements' to complete their task.

The bwareaopen function in MATLAB removes any pixels which are part of regions with a total area below the set threshold in a binary image. These regions are determined by identifying all pixels with a value of 1 that border another pixel with a value of 1 either vertically, horizontally, or diagonally. For segmentation algorithms, this removal of regions which fall outside the desired area can be extremely tempting; however, it must be used cautiously if the segmentation algorithm may encounter mutants that fall outside the realm of 'normal' and excluding phenotypically relevant segmentations is unacceptable.

Commands such as imdilate and imerode use structural elements to grow or shrink regions. They use the structural elements as a kernel to determine which pixel values need to be changed to complete the task. Structural elements range from disks to straight lines, among other shapes, and the choice of structural element is important as it affects the final shape of the region and sometimes erosions and dilations are not commutative (figure 4.2).

4.1.1.3 Image segmentation

Image segmentation comes from edge detection (including snake algorithms) and watershed transformations.

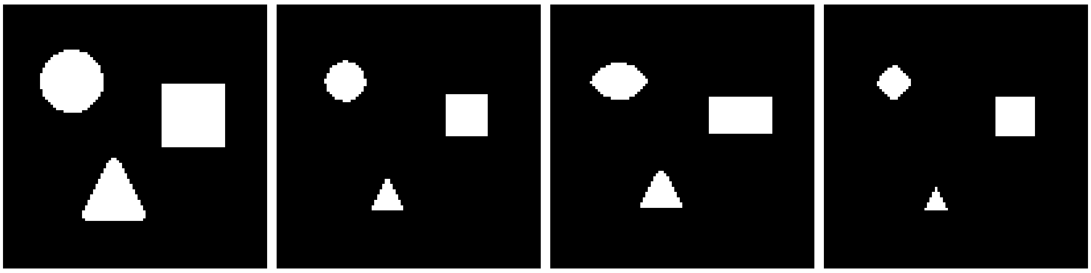


Figure 4.2: Examples of erosion by various structural element shapes: (a) input image; (b) horizontal line, width of 10 pixels; (c) disk, radius of 5 pixels; (d) square; length and width of 10 pixels.

Edge detection algorithms

Outside of the edge-emphasising filters mentioned in chapter 4.1.1.1 that identify edges, edge detection is commonly achieved via active contour models such as snakes. Snakes aim to create spline curves [105] by considering possible edge fragments and determining the minimum energy solution to construct a contiguous segment. They work extremely well in full colour images, where they have three channels to draw energy costs from, however, greyscale images only have one channel to analyse. This can reduce the effectiveness of active contour algorithms.

Additional constraints (such as gradient vector flow (GVF) [106]) can augment the algorithm's efficiency, on greyscale images. GVF works by weighting the energy cost based on the difference between neighbouring pixels intensities. Neighbouring pixels with similar intensities have a higher energy cost for active contour models, resulting in a segmentation that follows the edge with the largest contrast. This can be problematic if there are multiple segments in a frame because the number of objects to be found cannot be set in an automated algorithm and the overall lowest energy solution will merge segments if it is favourable. Both touching cells and boundaries with low contrast (inherent in transmitted light microscopy images or from poorly focussed images) increase the likelihood that segments are merged. This is undesirable in segmentation algorithms.

Watershed algorithms

The theory behind watershed transformations is that the greyscale image can be considered as a three-dimensional plot with the pixel values giving the data for the additional dimension. The larger the difference between neighbouring pixel values, the greater the drop between points. The edges of the regions of interest are usually visually identifiable by a sharp contrast that is continuous around the area. Such areas create a trough surrounded by a defined ridge or peak. The watershed transformation 'floods' the plain and the 'water' collects in the troughs, identifying the segment by filling it [107]. Watershed transformations can be optimised by using seeded watershed transformations among other methods [108, 109]. The seeds direct the 'water filling' so that the background is not also filled. This reduces the computational expense for the transformation.

4.1.1.4 Region analysis

Region analysis produces geometric properties of regions. This is usually the end goal for researchers using segmentation algorithms as it automates the analysis of the images. During the segmentation process, it is advisable to avoid/minimise using parameters/methods based on geometric properties as any cell which falls outside the declared expectation is lost from the analysis.

4.1.2 Existing software

4.1.2.1 PombeX

PombeX is the only automated segmentation algorithm designed specifically for the segmentation of fission yeast cells from microscopy images [62, 110]. While it is designed for use on transmitted illumination microscopy images, it can also

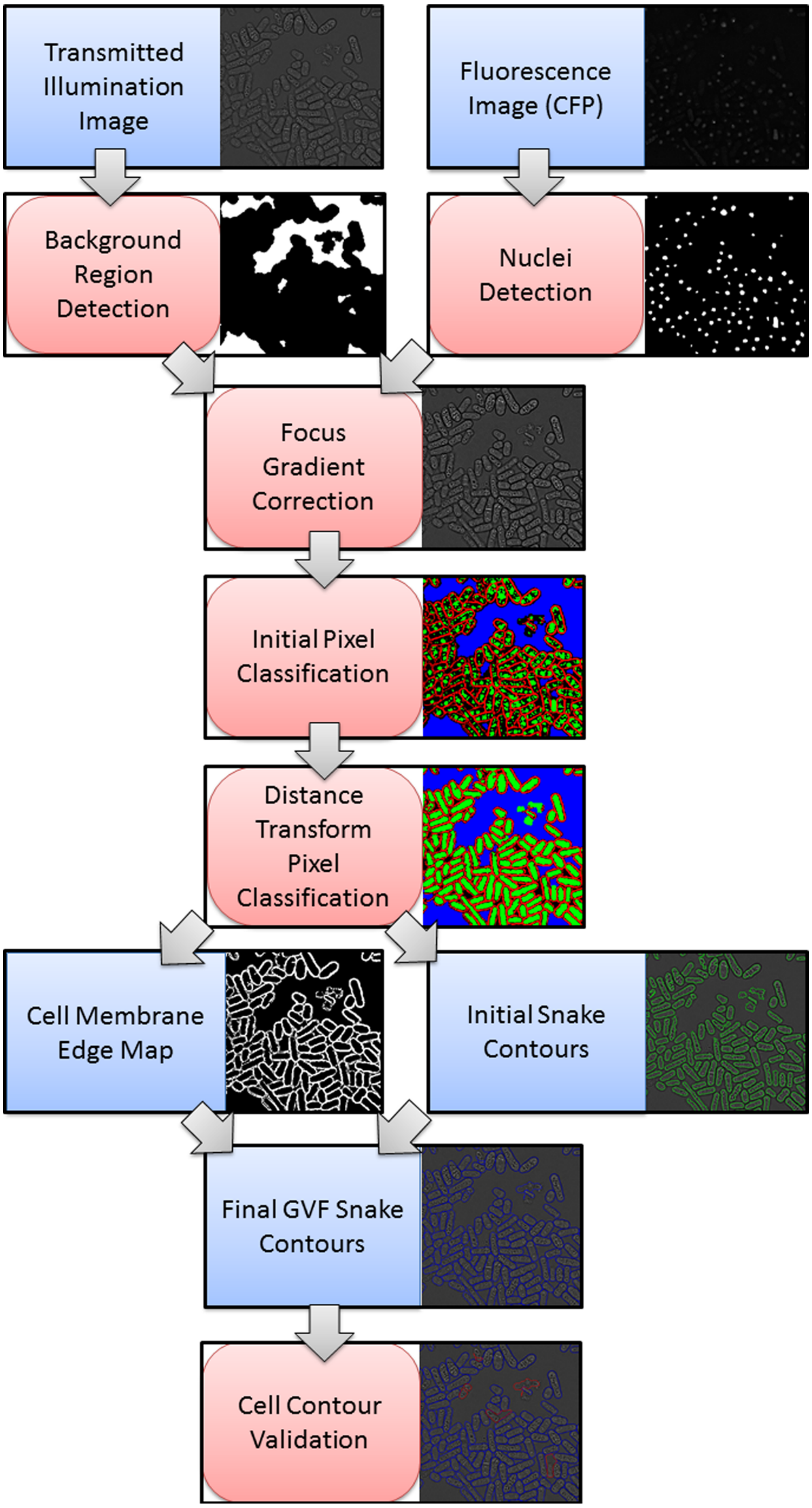

Transmitted Illumination Image
Fluorescence Image (CFP)
Background Region Detection
Nuclei Detection
Focus Gradient Correction
Initial Pixel Classification
Distance Transform Pixel Classification
Cell Membrane Edge Map
Initial Snake Contours
Final GVF Snake Contours
Cell Contour Validation

Figure 4.3: Schematic of PombeX program, taken from [62].

segment phase contrast microscopy images. There is also the option to use fluorescence microscopy images of fluorescently labelled nuclei to improve segmentation success. The PombeX pipeline is shown in figure 4.3.

Without the use of fluorescence microscopy images, PombeX detects the background region by using Otsu's method [98] on the shade corrected image. The focus gradient is then corrected producing an image which typically has a lighter inside of a cell and a darker background which is used for both the initial snake contour and the edge map for the gradient vector flow. Cell boundary contours are identified using the active contour model described in [105].

Adaptive thresholding using Otsu's method is used to classify the pixels as belonging to either the background, the inside of the cell, the cell membrane, or miscellaneous. The miscellaneously labelled pixels are reclassified using a distance-based method. The groups of pixels are then subjected to a series of morphological erosions and dilations to remove erroneously labelled pixels and smooth the regions. Now, the internal (from the equations given in both [62, 110]) and external (from the gradient vector flow snake from [106] based on the edge map) energies are used to improve the snake contours. The final snake contours are validated by machine learning classifiers which specifically target false positive segmentations.

When the fluorescence data is available, it is used to assist in the segmentation of the transmitted illumination images during several stages [62, 110]. Knowing the intensities of pixels guaranteed to be inside a cell means that the focus gradient correction and the threshold selected by Otsu's method for the classification of pixels inside the cell can be done in a more informed manner.

4.1.3 Aims and objectives

The aims and objectives of this chapter are:

- To develop a novel segmentation algorithm (which will be called Outline) primarily for the segmentation of *S. pombe* cells from phase contrast microscopy images.
- For Outline to perform better than PombeX, the available segmentation algorithm specifically designed for segmentation of *S. pombe* cells from transmitted light microscopy.
- To define any clear limitations of the novel segmentation algorithm based on imaging conditions.

4.2 Methodology

4.2.1 Previously described datasets and algorithms

This chapter uses the 100x-oil-Q, 100x-oil-Z, 60x-oil-Z, 60x-air-Z, and hybrid datasets described in chapter 2.1 along with the PombeX dataset [73]. The 100x-oil-Q and 100x-oil-Z datasets were used for the development of the Outline algorithm, and only the 100-oil-Z dataset was used to optimise the parameters.

The ground truth datasets for all of the above mentioned datasets are also used with the GTTest algorithm described in chapter 3.2.4.1 (except for the PombeX ground truth dataset which requires visual analysis using the criteria defined in chapter 2.5.1.

4.2.2 Iterations of Outline

There were three distinct versions of the novel segmentation algorithm which are of particular interest here: Ver1.0, Ver2.0, and Ver3.0. Each of these versions builds upon the results of the previous version as well as the findings of various other attempts and has adjustments to their code as described below.

4.2.2.1 Ver1.0

Ver1.0 enhances the contrast of the image and then filters it. The image is then subjected to morphological filtering before the cells are segmented (figure 4.4).

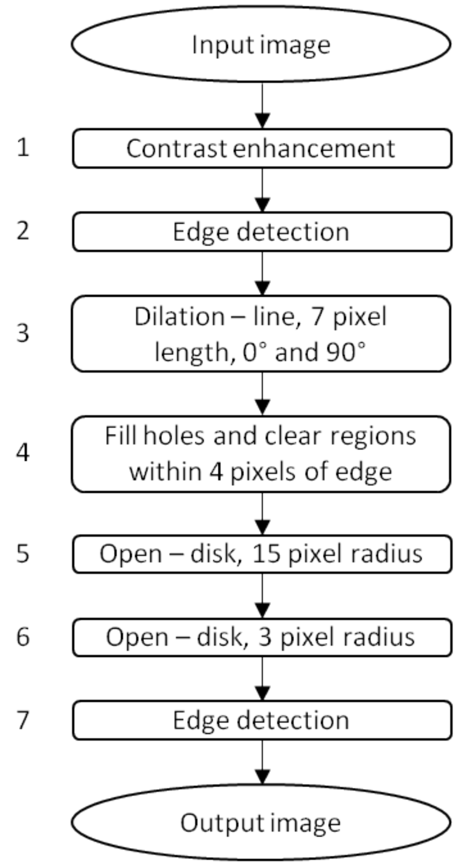

Input image
1
Contrast enhancement
2
Edge detection
3
Dilation – line, 7 pixel length, 0° and 90°
4
Fill holes and clear regions within 4 pixels of edge
5
Open – disk, 15 pixel radius
6
Open – disk, 3 pixel radius
7
Edge detection
Output image

Figure 4.4: Schematic of OutlineVer1.0.

The input image has its contrast maximised using imadjust to stretch the pixel intensities to cover the whole range of possible values and the edges detected with a LoG filter. Resultant lines are dilated using vertical and horizontal lines totaling 7 pixels in length (3 pixels each direction from the pixel of interest), and the regions filled. Segments within 4 pixels of the border are removed to ensure segmentations cannot include cells that potentially exceeds the frame. Remaining regions are then subjected to an erosion, followed by a dilation, using a disk with a radius of 15 pixels. The segments are then treated to another erosion followed by dilation, using a disk with a radius of 3 pixels. Each region's perimeter is found using bwperim. The edges are then dilated using a disk (1 pixel radius), coloured red, and overlaid on the contrast enhanced image to produce the output image.

Ver1.1 is based on Ver1.0 (figure 4.4) but the structural elements of steps 3, 5, and 6 have their length (3 pixels) or radius (15 pixels and 3 pixels) scaled based on total magnification and camera resolution.

4.2.2.2 Ver2.0

Ver2.0 (figure 4.5) differs from Ver1.0 by the addition of four extra steps inserted between steps 1 and 2 of Ver1.0 (figure 4.4).

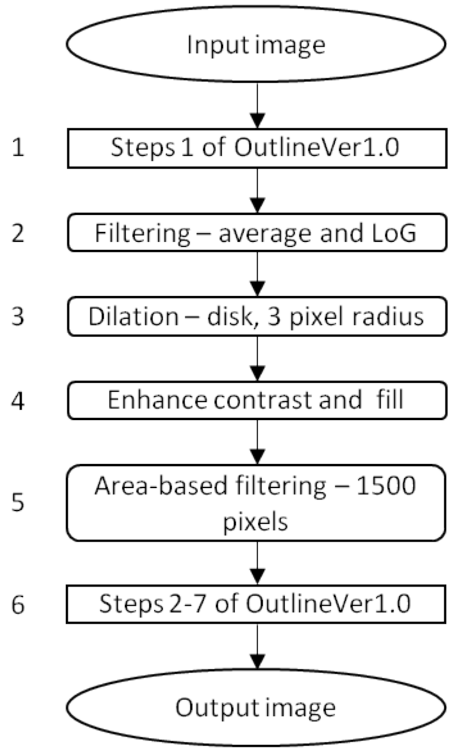

Input image
1
Steps 1 of OutlineVer1.0
2
Filtering – average and LoG
3
Dilation – disk, 3 pixel radius
4
Enhance contrast and fill
5
Area-based filtering – 1500 pixels
6
Steps 2-7 of OutlineVer1.0
Output image

Figure 4.5: Schematic of OutlineVer2.0.

The additional four steps begin with an averaging filter (with a 3 pixel by 3 pixel kernel) being used on the contrast maximised image, followed by a LoG filter with a 25 pixel by 25 pixel kernel. The potential segments are then dilated by a disk with a radius of 3 pixel. Next, the image is converted to a binary image using im2bw and a threshold of 0.5. Lastly, regions of less than 1,500 pixels are removed.

Ver2.1 and Ver2.2 are based on Ver2.0 (figure 4.5). Ver2.1 has the same scaling of parameters found in Ver1.1 (chapter 4.2.2.1) as well as the area-based filtering of step 5 (figure 4.5) being scaled. Ver2.2 has the same scaling of parameters as Ver2.1 and the contrast enhancement threshold of step 4 (figure 4.5) is set to 0.2.

4.2.2.3 Ver3.0

Investigations between Ver2.2 and Ver3.0 produced a different sequence of morphological filtering (figure 4.6). The first four steps of Ver2.2 are the same (chapter 4.2.2.2). A maximum area filter is added to step 5 of Ver2.0 (figure 4.5) and set to 10,000 pixel areas. Step 3 of Ver3.0 (figure 4.6) is the removal of regions within 4 pixels of the edge of the frame.

After these steps, Ver3.0 does not follow the pipeline of Ver1.0 or Ver2.0 until step 7 (figure 4.6) where the proposed segments are outlined and overlaid with the contrast maximised image. Instead, the new series of morphological filtering is: a disk (radius of 20 pixels) erosion; removal of regions with less than 300 pixels; and a dilation using a disk with a radius of 16 pixels (figure 4.6). All of these new parameters are also subjected to the scaling treatment of Ver1.1 and Ver2.1.

Ver3.1 is based on Ver3.0 with the only modification being the maximum area size in step 2 of Ver3.0 (figure 4.6) is increased from 10,000 to 50,000. Ver3.2 follows Ver3.1 adding a maximum area size to step 5 and subjects the output of step 3 through two different variations of step 4. The first variation is that of Ver3.1 which is used to make the final output image; the second variation is used to identify touching cells and remove the potential segmentations before step 7 of figure 4.6.

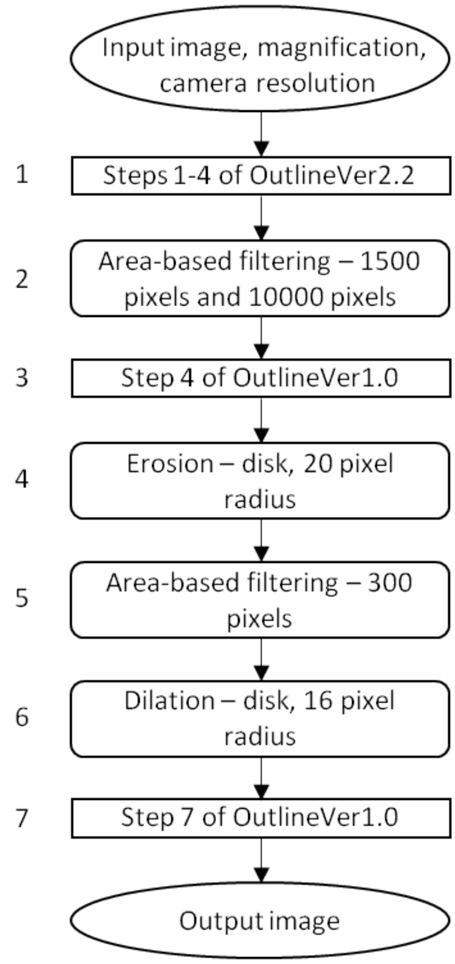

Input image, magnification, camera resolution
1
Steps 1-4 of OutlineVer2.2
2
Area-based filtering – 1500 pixels and 10000 pixels
3
Step 4 of OutlineVer1.0
4
Erosion – disk, 20 pixel radius
5
Area-based filtering – 300 pixels
6
Dilation – disk, 16 pixel radius
7
Step 7 of OutlineVer1.0
Output image

Figure 4.6: Schematic of OutlineVer3.0.

4.2.3 Parameter optimisation

Several parameters of Ver3.1 (figure 4.6) were identified for optimisation: the threshold used when using im2bw, the diameter of all disk structural elements (three in total), and the areas used by bwareaopen (three in total). The order of the parameter optimisation is shown in figure 4.7.

As the foreground-background threshold for im2bw is an independent variable it was explored first (figure 4.7).

The disk radii were the second group to be optimised (figure 4.7), as the areas of the regions for the filtering change depending on the size of the segments prior to the area filtering, which is dependent on the size of the structural elements. The first radius to be determined is that of the second dilation (step 6 in figure

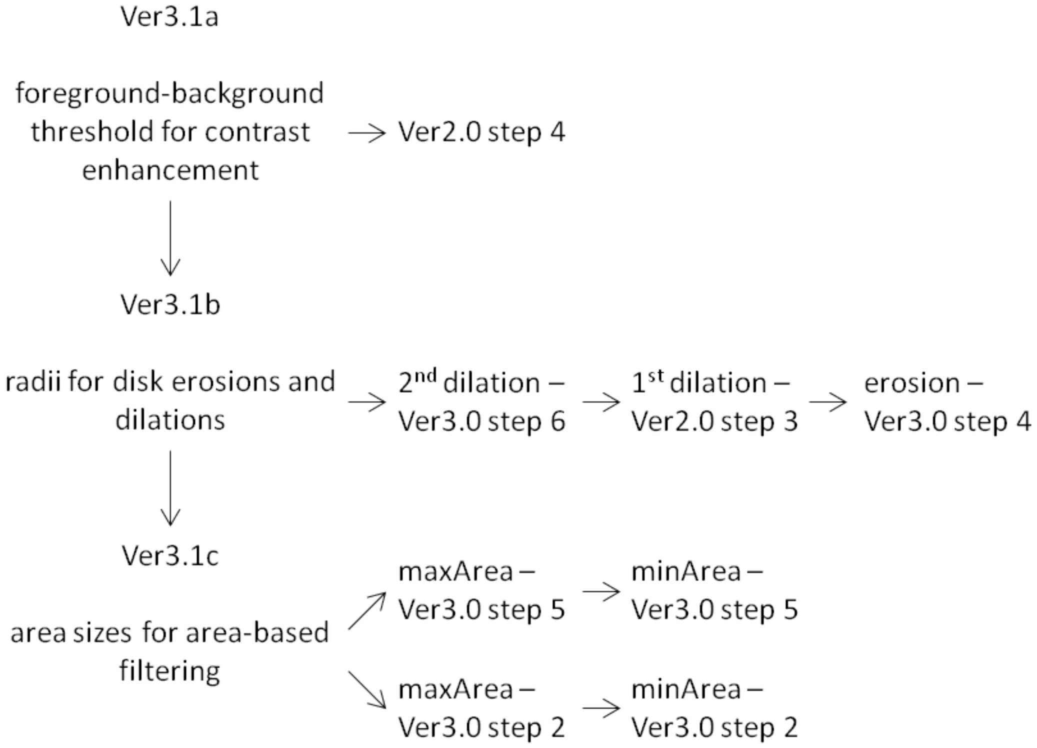

Ver3.1a
foreground-background threshold for contrast enhancement
Ver2.0 step 4
Ver3.1b
radii for disk erosions and dilations
2nd dilation – Ver3.0 step 6
1st dilation – Ver2.0 step 3
erosion – Ver3.0 step 4
Ver3.1c
area sizes for area-based filtering
maxArea – Ver3.0 step 5
minArea – Ver3.0 step 5
maxArea – Ver3.0 step 2
minArea – Ver3.0 step 2

Figure 4.7: Schematic for parameter optimisation.

4.6). This radius is not a set value, but rather described as an equation (equation 4.1). Currently, in Ver3.1, $c = 1$ (equations 4.2 and 4.3).

$$\text{second dilation radius} = \text{erosion radius} - \text{first dilation radius} - c \tag{4.1}$$

$$16 = 20 - 3 - c \tag{4.2}$$

$$c = 20 - 3 - 16 = 1 \tag{4.3}$$

The first dilation (step 3 in figure 4.5) is the second radius to be optimised as the erosion (step 4 in figure 4.6) depends on the size of the potential segments, which is dependent on the previous dilations/erosion. The radius for the first dilation was assessed over a range of 0 to 5 in intervals of 1. The radius for the erosion was assessed in intervals of 5 (from 0 to 35 for the erosion) and then at intervals of 1 for ±4 of the most promising value(s).

To assess the successfulness of erosions to separate touching cells, the area-based filters cannot remove touching cells. This is especially important as the areas used for the filtering have not been optimised and therefore can skew the result for the erosion radius. During the experiments of the erosion radius, the areas for the area-based filters are set to 0 (minimum value for step 2, and step 5, figure 4.6) and 1,050,000 (maximum value for step 2, figure 4.6).

Once the disk radii are determined, the region sizes for the various area-based filtering steps can be optimised (figure 4.7). There are three area sizes to be determined: the minimum area pre-erosion (step 2, figure 4.6), the maximum area pre-erosion (step 2, figure 4.6), and the minimum area post-erosion (step 5, figure 4.6). Optimising the areas is dependent on the size of potential segmentations at the relevant stage of the algorithm, and therefore, can only be completed once the structural elements for the erosion and dilations have been set.

As with the disk radius for the erosion, the areas in pixels are examined in large intervals before smaller intervals are assessed based on previous results. The smaller intervals for 'fine-tuning' will be 10% of the large intervals. The minimum area pre-erosion will initially be explored in intervals of 1,000, while for the maximum area will be of 10,000. For the post-erosion area filtering, the large intervals will be of 1,000 for the minimum area and 10,000 for the maximum. All initial intervals will begin at 0 and end at twice the value given in Ver3.1 (chapter 4.2.2.3) unless the $F_{0.5}$ scores have not appeared to have peaked. If this is the case, the larger intervals will continue until the $F_{0.5}$ scores for three consecutive intervals show a post-peak decreasing trend. Pre-erosion filters are explored using Ver3.1 while post-erosion filters are examined using Ver3.2.

The final iteration of the segmentation algorithm is depicited in figure 4.8.

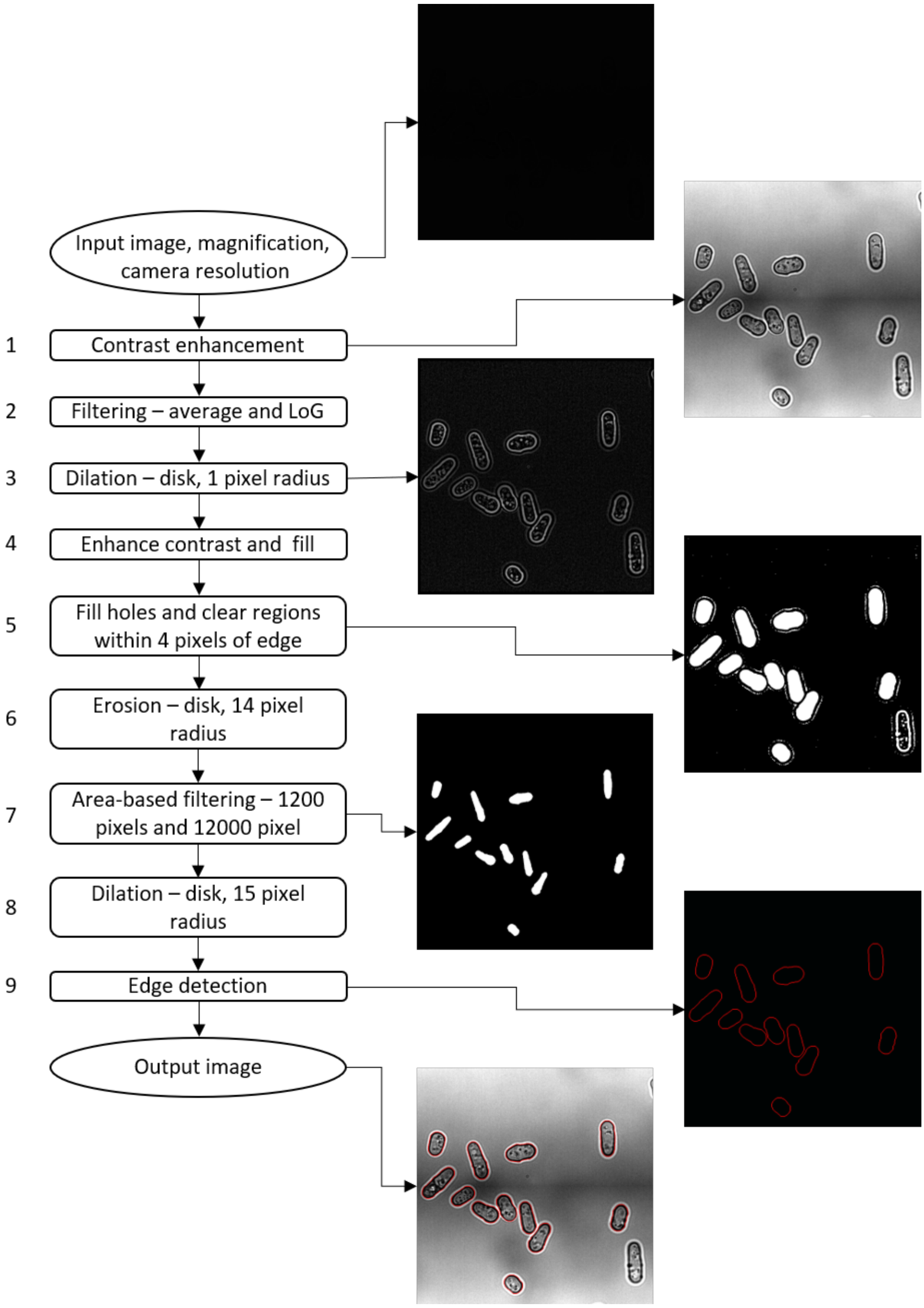

Input image, magnification, camera resolution
1
Contrast enhancement
2
Filtering – average and LoG
3
Dilation – disk, 1 pixel radius
4
Enhance contrast and fill
5
Fill holes and clear regions within 4 pixels of edge
6
Erosion – disk, 14 pixel radius
7
Area-based filtering – 1200 pixels and 12000 pixel
8
Dilation – disk, 15 pixel radius
9
Edge detection
Output image

Figure 4.8: Schematic of the optimised Outline algorithm. Example output images provided for key steps.

4.2.4 Exploring usable imaging conditions based limitations of Outline

Once the parameters were optimised for the 100x-oil-Z dataset, the novel segmentation algorithm (Outline) was run on all of the datasets mentioned in chapter 4.2.1. The results were then analysed to identify any limitations of the segmentation algorithm resultant from imaging conditions.

4.2.5 Comparing segmentation algorithm success to PombeX

Three combinations of segmentation algorithms and datasets are used to compare the novel segmentation algorithm to the PombeX segmentation algorithm. First, the PombeX segmentation algorithm was used with their dataset to provide a baseline for comparison using the manual assessment method outlined in chapter 2.5.1. Second, the PombeX program is used on both the 100x-oil-Z and the hybrid datasets. Finally, the novel segmentation algorithm is used with the PombeX dataset.

4.3 Results and Discussion

4.3.1 Development of a novel segmentation algorithm

The development of the novel segmentation algorithm, Outline, resulted in numerous versions being trialled. The total number of true positives, false positives, and false negatives, as well as the $F_{0.5}$ scores for J values of 0.75-0.95 (in increments of 0.05) are shown in table 4.2 (count of cell/segment classification) and table 4.1 ($F_{0.5}$ scores) for both datasets.

		J value				
	Iteration	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
100x-oil-Q	Ver 1.0	0.0462	0.0071	0.0000	0.0000	0.0000
	Ver 1.1	0.3168	0.1863	0.1522	0.0901	0.0031
	Ver 2.0	0.3683	0.2456	0.1023	0.0102	0.0000
	Ver 2.1	0.6360	0.6320	0.5960	0.4200	0.0320
	Ver 2.2	0.6802	0.6625	0.6331	0.5153	0.0530
	Ver 3.0	0.3821	0.3738	0.3322	0.1744	0.0000
	Ver 3.1	0.7266	0.6999	0.6257	0.3470	0.0148
100x-oil-Z	Ver 1.0	0.1532	0.0486	0.0224	0.0137	0.0050
	Ver 1.1	0.1806	0.0648	0.0249	0.0137	0.0037
	Ver 2.0	0.5998	0.5941	0.5344	0.1961	0.0199
	Ver 2.1	0.5998	0.5998	0.5628	0.2757	0.0256
	Ver 2.2	0.6353	0.5794	0.4219	0.1995	0.0165
	Ver 3.0	0.8295	0.7263	0.5195	0.2347	0.0125
	Ver 3.1	0.8728	0.7825	0.5556	0.2370	0.0113

Table 4.1: Table of $F_{0.5}$ scores for Outline development for the 100x-oil-Q and 100x-oil-Z datasets over a range of J values.

4.3.1.1 Initial segmentation algorithm

Ver1.0 (described in chapter 4.2.2.1) is the initial pipeline designed for this novel segmentation algorithm and was used to iterate upon.

Some datasets consist of images that appear as black squares when opened outside of MetaMorph to the observer (figure 4.9a). This means that the first step of Ver1.0 is to maximise the contrast of the image using imadjust (figure 4.9b). Contrast enhancement is important even if the images are already clearly visible (figure 4.9c) as it maximises the contrast which is helpful for subsequent steps (figure 4.9d).

4.3.1.2 Image enhancement by filtering

With the additional filtering of Ver2.0, the algorithm gives higher $F_{0.5}$ scores on both datasets (table 4.1). This indicates that the extra steps (steps 2-5 in figure 4.5) benefit the segmentation algorithm.

		J value														
		0.75			**0.80**			**0.85**			**0.90**			**0.95**		
	Iteration	**tp**	**fp**	**fn**	**tp**	**fp**	**fn**	**tp**	**fp**	**fn**	**tp**	**fp**	**fn**	**tp**	**fp**	**fn**
100x-oil-Q	Ver 1.0	13	238	389	2	249	400	0	251	402	0	251	402	0	251	402
	Ver 1.1	102	200	300	60	242	342	49	253	353	29	273	373	1	301	401
	Ver 2.0	108	158	294	72	194	330	30	236	372	3	263	399	0	266	402
	Ver 2.1	159	53	243	158	54	244	149	63	253	105	107	297	8	204	394
	Ver 2.2	231	93	171	225	99	177	215	109	187	175	149	227	18	306	384
	Ver 3.0	46	4	356	45	5	357	40	10	362	21	29	381	0	50	402
	Ver 3.1	245	76	157	236	85	166	211	110	191	117	204	285	5	316	397
100x-oil-Z	Ver 1.0	123	659	764	39	743	848	18	764	869	11	771	876	4	778	883
	Ver 1.1	145	637	742	52	730	835	20	762	867	11	771	876	3	779	884
	Ver 2.0	211	7	676	209	9	678	188	30	699	69	149	818	7	211	880
	Ver 2.1	211	7	676	211	7	676	198	20	689	97	121	790	9	209	878
	Ver 2.2	500	262	387	456	306	431	332	430	555	157	605	730	13	749	874
	Ver 3.0	463	13	424	411	65	476	290	186	597	131	345	756	7	469	880
	Ver 3.1	696	79	191	624	151	263	443	332	444	189	586	698	9	766	878

Table 4.2: Table of total number of true positive (tp), false positive (fp), and false negative (fn) segmentations for Outline development for the 100x-oil-Q and 100x-oil-Z datasets over a range of J values.

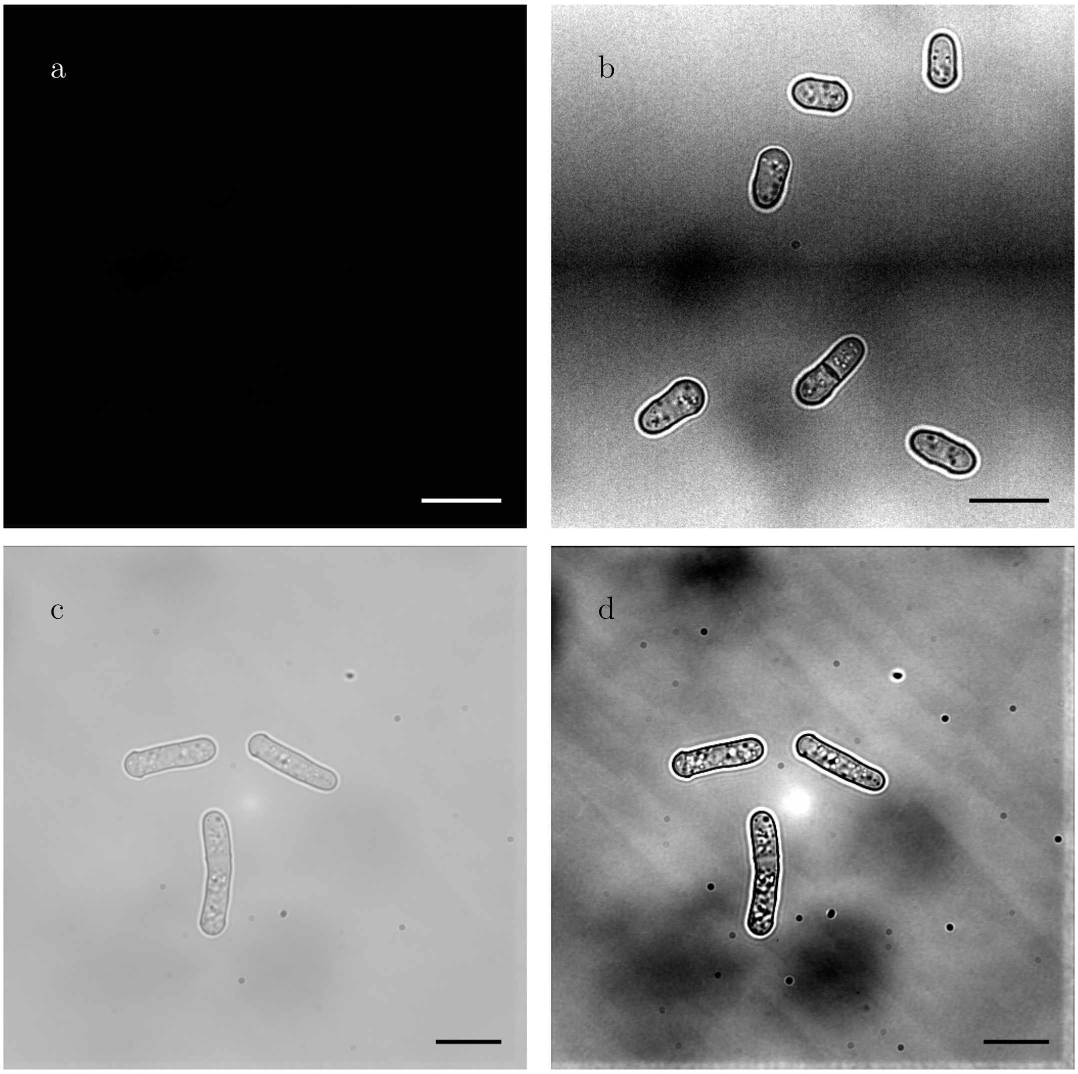


Figure 4.9: Effect of imadjust function. (a) input and (b) output of the 100x-oil-Z dataset. (c) input and (d) ouput of the 100x-oil-Q dataset. Scale bars represent 10 µm.

An averaging filter and a LoG filter was chosen over an adaptive filter (as mentioned in chapter 4.1.1.1) for step 2 of figure 4.5. This is because the contrast of the cell walls requires a little blurring to ensure that a single, clear edge is detected before the morphological filtering occurs; however, an adaptive filter produced a poor estimation of the desired output. It is likely due to the fact that the cell wall is not considered particularly thick by the function and the decreased blurring does not allow for the smoothing of the different levels of contrast found around the cell.

4.3.1.3 Scaling parameters based on imaging conditions

Ver2.0 performs consistently poorer on the 100x-oil-Q dataset than the 100x-oil-Z dataset, which is comparable to the performance of Ver1.0 (table 4.1). The only difference between the datasets is the camera used and their resolutions. This suggests that the size of structural elements may be dependent on imaging conditions. Even more importantly, the pixel count for a region varies dramatically when the magnification and resolution is taken into account; for example, a 100 pixel area represents 256 µm^2 in the 100x-oil-Q dataset but only 42.25 µm^2 in the 100x-oil-Z dataset. This means that the resolution and the magnification need to be taken into account in the scaling of the parameters.

To address this issue, all structural elements for dilations and erosions are scaled accordingly by dividing by the scaling factor (equation 4.4) of the 100x-oil-Z dataset, then multiplying this to-be-scaled value by the scaling factor of the dataset the algorithm is analysing, before the final value is rounded to the nearest whole number. All areas used for region-based filtering undergo this scaling twice to account for the additional dimension. The values are first divided by the scaling factor of the 100x-oil-Z dataset as the values were originally determined for that dataset via trial and error. This division calculates the to-be-scaled value for the parameter.

$$\text{scaling factor} = \frac{\text{total magnification}}{\text{camera resolution}} \tag{4.4}$$

Scaling of the parameters in Ver1.0 and Ver2.0 result in Ver1.1 and Ver2.1. As pre-scaled parameter value is determined using the reciprocal of the 100x-oil-Z dataset, it is expected that Ver1.1 and Ver2.1 would perform in the exact same way as Ver1.0 and Ver2.0 respectively. From table 4.1 we can see that this is not the case. Table 4.2 shows that the total number of proposed segmentations is the same for each pair of alogrithms, but some segments are classified differently between the non-scaled and scaled iterations. This suggests that the final segmentations have slight variances caused by the scaling which alters the J score minutely for individual segments.

On the 100x-oil-Q dataset, Ver1.1 identifies 61 extra proposed segmentations compared to Ver1.0 (table 4.2). Not only are more cells correctly segmented (shown by the increase in true positive segmentation for all J values, table 4.2), but the accuracy of these segmentations has also improved (table 4.1). The total number of proposed segmentations decreases when comparing Ver2.0 to Ver2.1 (table 4.2); however, the $F_{0.5}$ scores all show an improvement of Ver2.1 (table 4.1), which means that the lost proposed segmentations are false negatives that would not improve with parameter optimisation.

4.3.1.4 Changing the foreground-background threshold

Even though Ver2.1 produces higher $F_{0.5}$ scores compared to Ver1.1 (table 4.1), the total number of proposed segmentations is dramatically lower over both datasets (430 compared to 1,084, table 4.2). In an attempt to increase the number of proposed segmentations, the value selected for im2bw was lowered from the default of 0.5 to 0.2 in Ver2.1. This modification resulted in Ver2.2.

Changing the threshold for foreground-background separation to 0.2 increases the total number of proposed segments. Ver2.2 produces 1086 proposed segmentations over both datasets which is more than the 1084 proposed segments of Ver1.1 (table 4.2). This suggests that the foreground-background threshold parameter should be further explored during the optimisation stage.

4.3.1.5 Separating touching cells via erosion

Visual analysis of Ver2.2 shows that the predominant cause of false positive segmentations was of touching cells being segmented together as one cell. While several methods were trialled, few showed promising results.

Eroding potential segments in an attempt to cause touching cells to separate produced the most consistent results; however, cells aligned as in figure 1.8c can never be tidily separated via this method because the touching plane erodes and results in a misshapened segment to be dilated (figure 4.10l). Other orientations of touching cells will be separated as long as the erosion exceeds the width of the of the touching plane (figure 4.10).

Cells orientated as in figure 4.10i can be excluded from analysis without skewing results as there is no phenotypical reason for this alignment. While the width of these segments is double that of singular cells, isolating them for further processing is inadvisable. This is because of how regionprops measures widths (explained in chapter 4.1.1.4). If the cell is bent, the optimal fitted ellipse can cause false width measurements which means a maximum width measurement cannot be used as a meaningful filter. This leaves area-based filtering as a method to attempt to remove inappropriate segments.

Two extra area-based filters are introduced between Ver2.2 and Ver3.0 (steps 2 and 5, figure 4.6). On the 100x-oil-Z dataset, Ver3.0 outperforms Ver2.2 while J < 0.95 (table 4.1); however, Ver2.2 performs better on the 100x-oil-Q dataset (table 4.1). The reason for this is likely tied to the fact that Ver2.2 found over six times the number of proposed segmentations compared to Ver3.0 on the 100x-oil-Q dataset (table 4.2). This causes a large number of false negatives, regardless of how many segmentations are true positives, which limits the $F_{0.5}$ score. The total number of proposed segmentations for Ver2.2 on the 100x-oil-Z dataset is less than 1.5 times that of Ver3.0 (table 4.2), meaning that is is less limiting than for the 100x-oil-Q dataset.

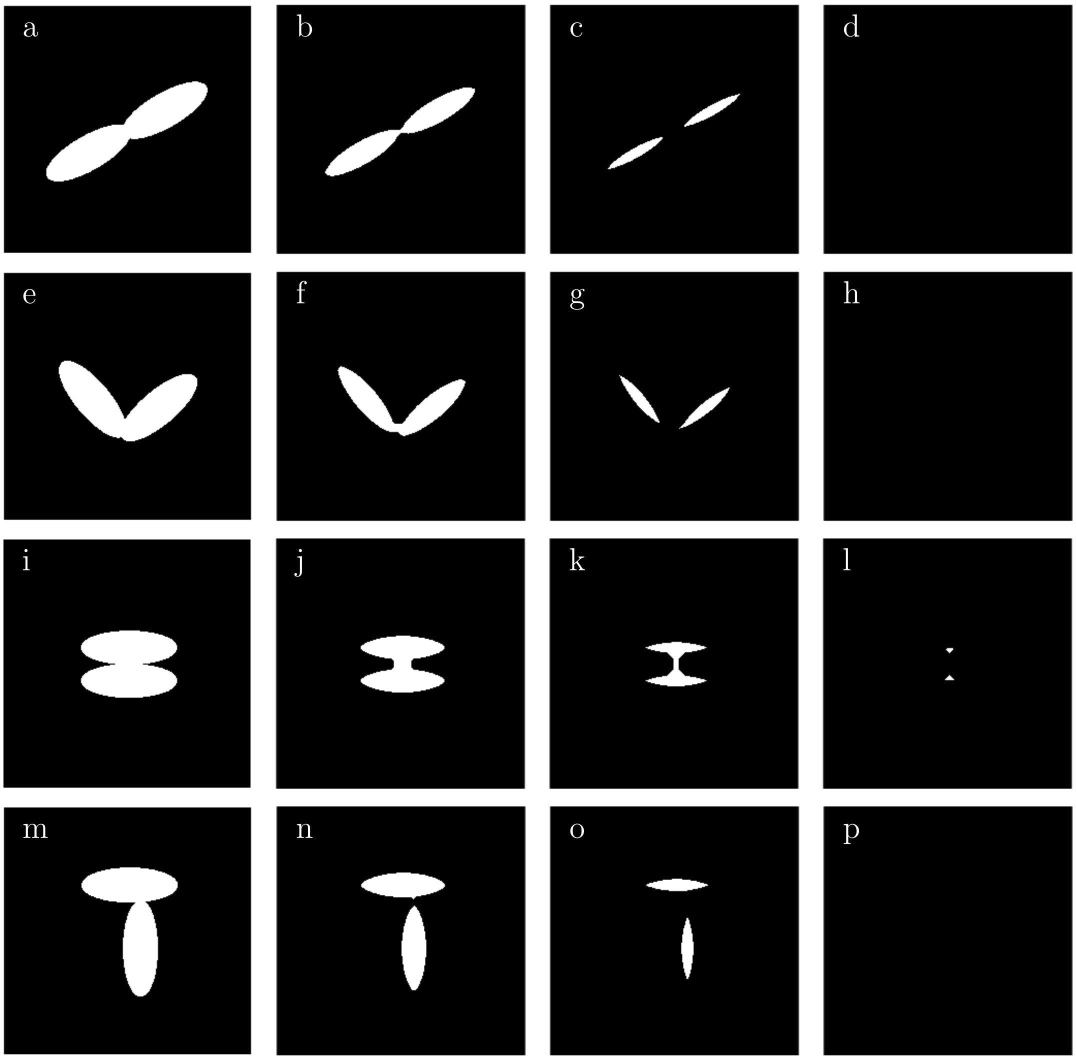


Figure 4.10: Illustration of separating touching cells by erosion using a disk. (L)-(R): input image; erosion of input, disk radius of 6 pixels; erosion of input, disk radius 12 pixels; erosion of input, disk radius 18 pixels.

4.3.1.6 Changing the maximum possible region area

Visual analysis of output images from Ver3.0 shows that many promising segmentations from Ver2.2 are not present. All of these cells are reasonably large so it was hypothesised that the maximum potential segment size used in the filtering of step 2 of figure 4.6 caused them to be removed from consideration. The maximum region size was increase to 50,000 (Ver3.1, chapter 4.2.2.3).

Ver3.1 produces higher $F_{0.5}$ scores for both datasets, except for the 100x-oil-Z dataset when J is 0.95 (table 4.1). On the 100x-oil-Q dataset, Ver3.1 increases the total number proposed segmentations from 50 to 321 (table 4.2). The increase in the $F_{0.5}$ scores proves that the majority of these extra proposed segmentations are not detrimental to the functionality of the segmentation algorithm.

4.3.2 Parameter optimisation

The $F_{0.5}$ scores for Ver3.1 are above 0.700 for both of the 100x-oil-Q and 100x-oil-Z datasets while J is 0.75 (table 4.1). The decreases in the $F_{0.5}$ scores as the value of J increases suggests that the quality of the segmentations is the limiting factor. This means that the parameters of Ver3.1 need to be optimised to increase the $F_{0.5}$ scores at higher J values.

4.3.2.1 Segmentation quality assessment

During the optimisation of parameters, the use of ground truth segmentations to assess the quality of proposed segmentations may not always be the most informative option. This is because a perfectly shaped, but incorrectly scaled, segmentation is more valuable during parameter optimisation than a grossly misshapen segment, especially when considering the use of morphological filters in this novel algorithm. Because of this, optimisation results do not always depend on comparisons to the ground truth segmentations.

4.3.2.2 Foreground-background threshold

Varying the threshold value used for separating the foreground from background can be analysed based on comparisons to ground truth segmentations. This is because the size and shape of the proposed segmentations only change based on which pixels are labelled as foreground pixels, so their quality varies minimally between threshold values.

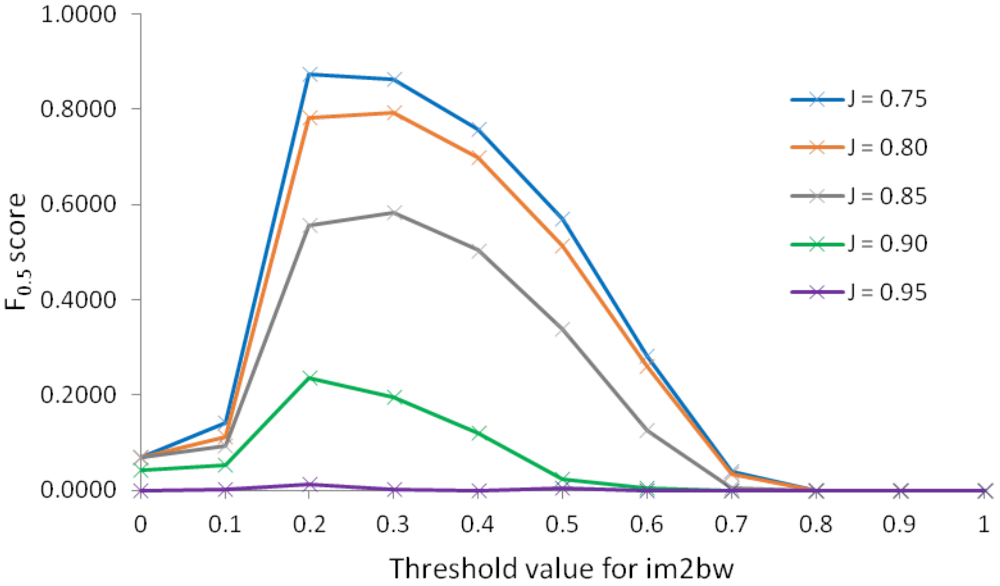


Figure 4.11: Graph of $F_{0.5}$ scores for the optimisation of the foreground-background threshold parameter over a range of J values.

Otsu's method [98] was used to determine the threshold value based on the output image from step 3 (figure 4.5). Over all J values, Otsu's performed worse than a consistently set value of 0.20 and 0.30 (excluding when J is 0.95 for the latter; table 4.3). In theory, selecting a value based on the pixel intensities of each frame should perform optimally, however, the varity of pixel intensities within the background noise results in background pixels becoming foreground pixels to minimise the 'within-class variance' [98].

		J value									
Threshold		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
value	**PS**	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$
0.00	13	13	0.0692	13	0.0692	13	0.0692	8	0.0426	0	0.0000
0.10	262	55	0.1421	43	0.1111	36	0.0930	20	0.0517	1	0.0026
0.20	775	696	0.8728	624	0.7825	443	0.5556	189	0.2370	9	0.0113
0.30	542	528	0.8642	484	0.7921	346	0.5827	120	0.1964	1	0.0016
0.40	364	355	0.7576	328	0.7000	236	0.5036	56	0.1195	0	0.0000
0.50	199	192	0.5704	173	0.5140	114	0.3387	8	0.0238	1	0.0030
0.60	66	65	0.2824	60	0.2606	29	0.1260	1	0.0043	0	0.0000
0.70	7	7	0.0383	6	0.0328	1	0.0055	0	0.0000	0	0.0000
0.80	0	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
0.90	0	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
1.00	0	0	0.0000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
Otsu's	645	530	0.7643	480	0.6922	347	0.5004	128	0.1846	2	0.0029

Table 4.3: Table of $F_{0.5}$ scores and number of true positive (tp) segmentations for the optimisation of the foreground-background threshold parameter over a range of J values. The number of proposed segmentations (PS) is also given.

Multi-level thresholding is also not suitable as it is not possible to predict the number of levels required to successfully isolate the true foreground pixels. Estimating their quantity is dependent on the number of cells in the frame and the imaging conditions. While the magnification and resolution of the imaging conditions are known, the range which the intensities cover is not. This range affects the mean and, in turn, the variance. The number of cells in an image would affect the proportion of pixels that are anticipated as foreground pixels because more cells increases the expected number of foreground pixels.

This means a predetermined, set threshold appears to be favourable. A threshold of 0.20 results in the highest $F_{0.5}$ scores while J is less than 0.80 or greater than 0.85; when J is 0.80 or 0.85, a threshold of 0.30 performs better (figure 4.11, table 4.3). The higher $F_{0.5}$ scores for a threshold of 0.30 compared to 0.20 at certain J values indicates that the quality of the proposed segmentations between the two thresholds varies; however, the quality of the proposed segmentations can likely be improved during the morphological filtering stages. Because of this, a foreground-background threshold level of 0.20 was selected for this novel segmentation algorithm.

4.3.2.3 Disk radius for the second dilation of proposed segmentations

The first radius to set is the that for the second dilation (figure 4.7). As discussed in chapter 4.2.3, this radius is covered by equation 4.1; however, the optimal value for c needs to be determined.

Setting this parameter is a quality based problem. This means that the higher J values are more informative than the lower ones as they indicate better proposed segmentations. Table 4.4 shows that a c value of +1 and 0 produce the highest $F_{0.5}$ scores when J is 0.75 and 0.80 respectively. When J $\geq$ 0.85, a c value of -2 is consistently best (table 4.4). The higher $F_{0.5}$ scores for $c = 0$ and $c = +1$ indicate that a larger proportion of the proposed segmentations are deemed true

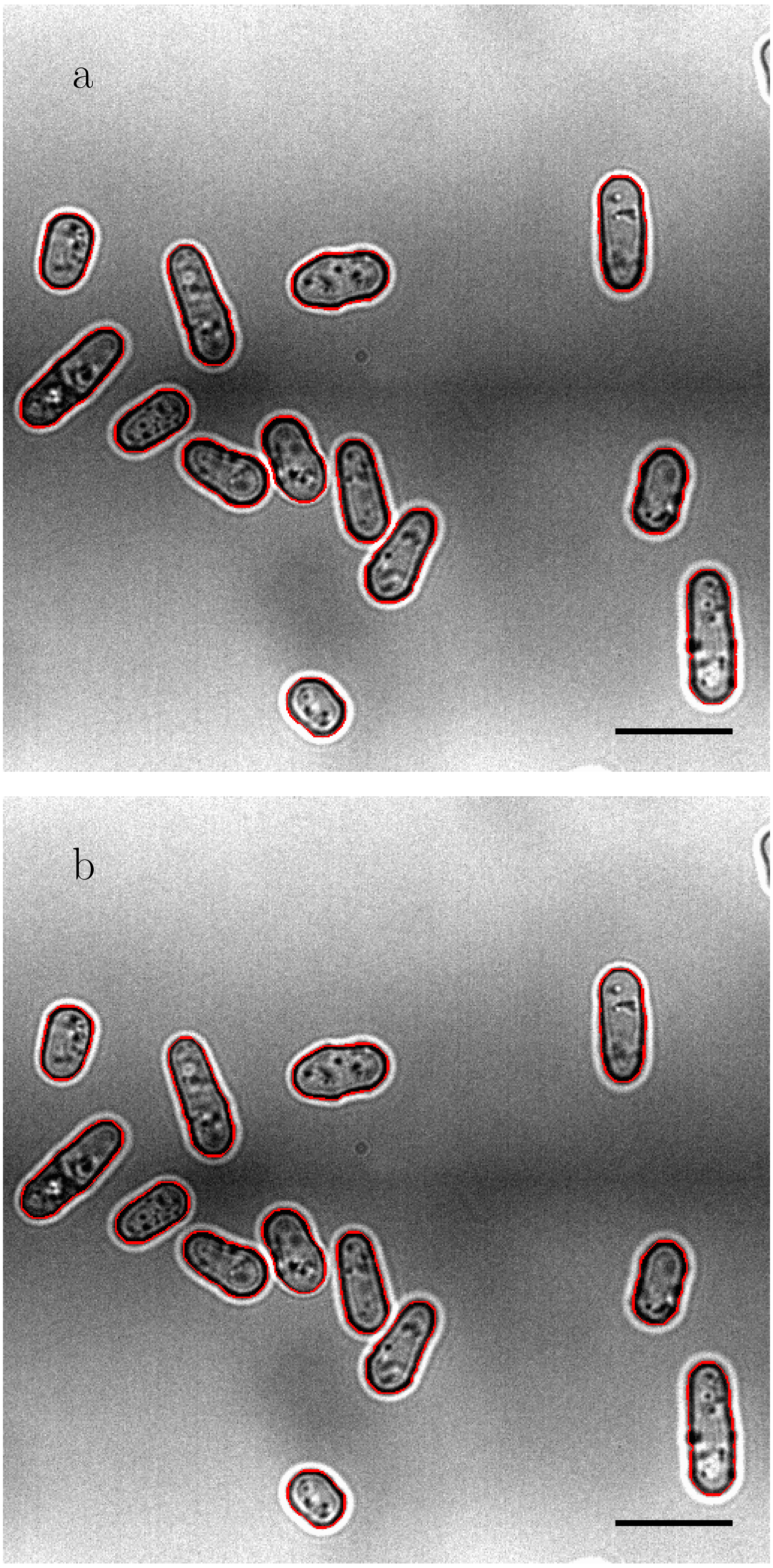


Figure 4.12: Example output image for different c values: (a) $c =$ -2, (b) $c =$ -1, (c) $c = 0$, (d) $c = +1$. Proposed segmentations shown in red. Scale bars represent 10 µm.

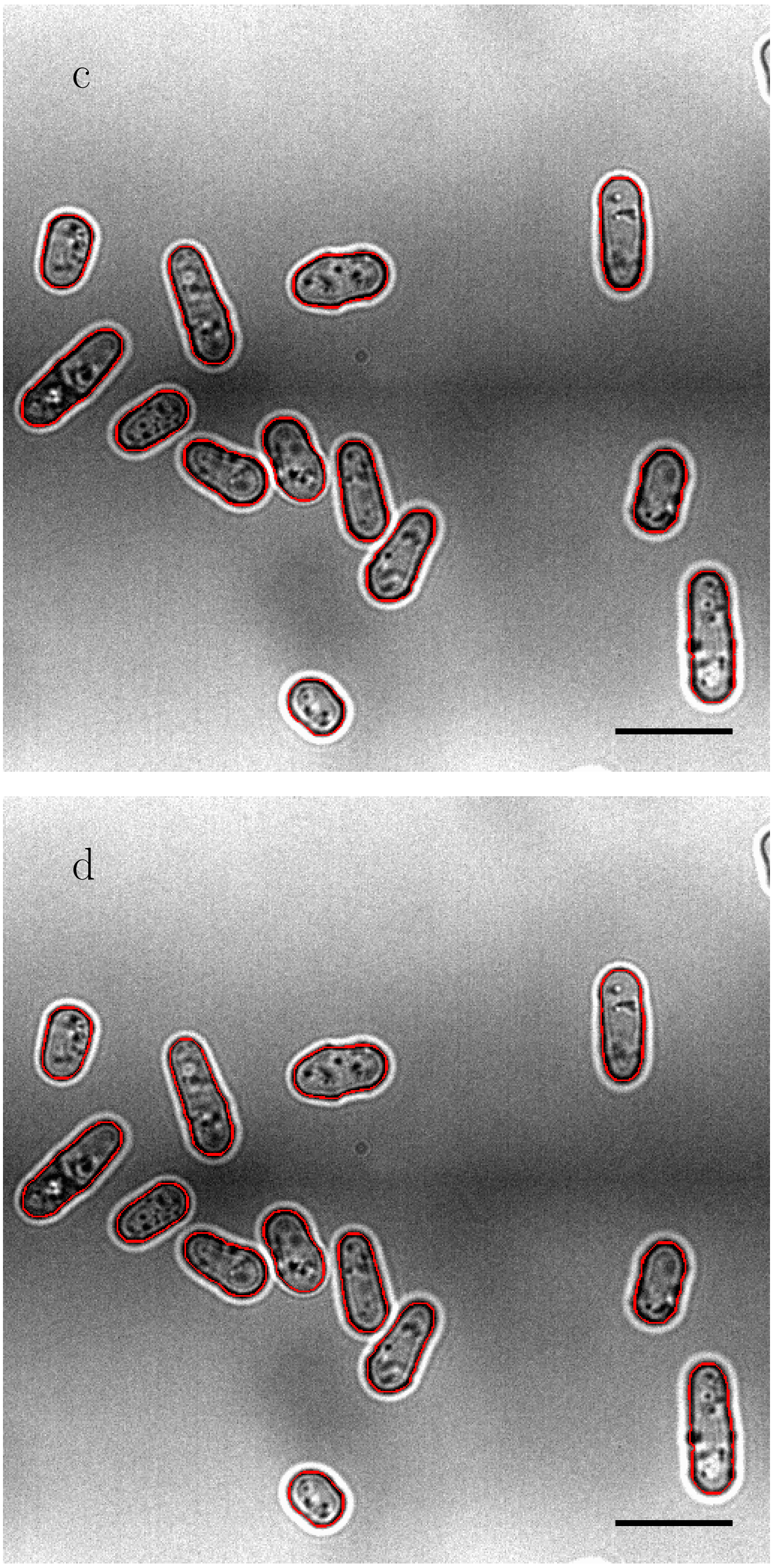


Figure 4.12: Example output image for different c values: (a) $c =$ -2, (b) $c =$ -1, (c) $c = 0$, (d) $c = +1$. Proposed segmentations shown in red. Scale bars represent 10 µm.

	J value				
***c* value**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
-3	0.8352	0.8189	0.7637	0.5242	0.0953
-2	0.8427	0.8302	0.7988	0.7223	0.1480
-1	0.8540	0.8327	0.7938	0.6270	0.1279
0	0.8565	0.8340	0.7512	0.5142	0.0803
+1	0.8728	0.7825	0.5556	0.2370	0.0113
+2	0.8252	0.5932	0.2445	0.0314	0.0000
+3	0.7537	0.4791	0.1204	0.0201	0.0000

Table 4.4: Table of $F_{0.5}$ scores for the optimisation of the radius for the second dilation parameter over a range of J values. The c value is the constant difference between the first dilation radius and the erosion radius (defined by equation 4.1).

positives; however, difference in $F_{0.5}$ scores when J is 0.80 is negligible (0.8340, 0.8327, and 0.8302 when c is 0, -1, and -2 respectively; table 4.4) and, other than when J is 0.75, when $c = 1$ produces lower $F_{0.5}$ scores than when c is between 0 and -3.

As this experiment is about improving the quality of the proposed segmentations, the higher J values are deemed more informative. The visual analysis confirms that when $c = -2$, the quality of the segments improves overall as the proposed segmentations delineate the outer edge of the cell wall. Figure 4.12 shows the minuscule differences in placement of the proposed segmentation along the cell wall. This supports the disk radius for the second dilation being expressed by equation 4.5.

$$\text{final dilation radius} = \text{erosion radius} - \text{first dilation radius} + 2 \tag{4.5}$$

4.3.2.4 Disk radius for the first dilation of proposed segmentations

The first dilation is explored as outlined in figure 4.7. The $F_{0.5}$ scores are shown in table 4.5.

First						**J value**					
dilation		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
radius	**PS**	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$
0	617	607	0.9046	603	0.8987	599	0.8927	543	0.8092	94	0.1401
1	685	672	0.9264	665	0.9167	655	0.9030	559	0.7706	102	0.1406
2	737	699	0.9113	694	0.9048	684	0.8957	620	0.8083	125	0.1630
3	775	672	0.8427	662	0.8302	637	0.7988	576	0.7223	118	0.1480
4	782	578	0.7198	538	0.6700	501	0.6239	396	0.4932	83	0.1034
5	720	236	0.3132	186	0.2469	153	0.2031	106	0.1407	9	0.0119

Table 4.5: Table of $F_{0.5}$ scores and number of true positive (tp) segmentations for the optimisation of the first dilation parameter over a range of J values. The number of proposed segmentations (PS) is also given. Radii given in pixels.

A radius of 1 for the first dilation provides the highest $F_{0.5}$ scores while J $\leq$ 0.85 (table 4.5). When J is 0.90 or 0.95, radii of 0 and 2 (respectively) produce the highest $F_{0.5}$ scores (table 4.5). This variation, especially at the highest three J values, makes it less clear as to which radius to use.

Unlike with the second dilation radius, the number of proposed segmentations changes as the proposed radius changes (table 4.5). This suggests that the change in radius does not affect the overall quality of a proposed segmentation, rather, if the potential segmentation is removed before the end of the pipeline. This idea is supported by figure 4.13 as an extra proposed segmentation appears in each frame as the radius increases but the quality of the proposed segmentations remain fairly consistent.

The difference in the total number of proposed segmentations is caused by the removal of potential segmentations at the area-based filtering stage (step 2, figure 4.6). The probability of complete perimeters of potential segmentations with small dilation radii decreases, which in turn does not permit the region to be filled (step 4, figure 4.5). Without complete filling, potential segmentations do not meet the minimum region area. The difference in perimeter completeness depending on dilation radius is demonstrated in figure 4.14.

The slight variations in proposed segmentation quality can be seen in figure 4.15. Only the proposed segmentation in the green box of figure 4.15c is visually different from those in figures 4.15a-b; however, figures 4.15d-f show the pixels included in one version of proposed segmentations but not the other it is compared to. These small differences can have a large affect on the $F_{0.5}$ score at the higher J values.

Choosing between a radius of 0, 1, or 2 is not straight forward. Using the $F_{0.5}$ scores for when J $\geq$ 0.90 (the mean J value for the dataset; table 3.1) suggests a radius of either 0 or 2 (table 4.5), while a radius of 1 (the mean of 0 and 2) provides the best $F_{0.5}$ scores for all other J values (table 4.5). It is also interesting

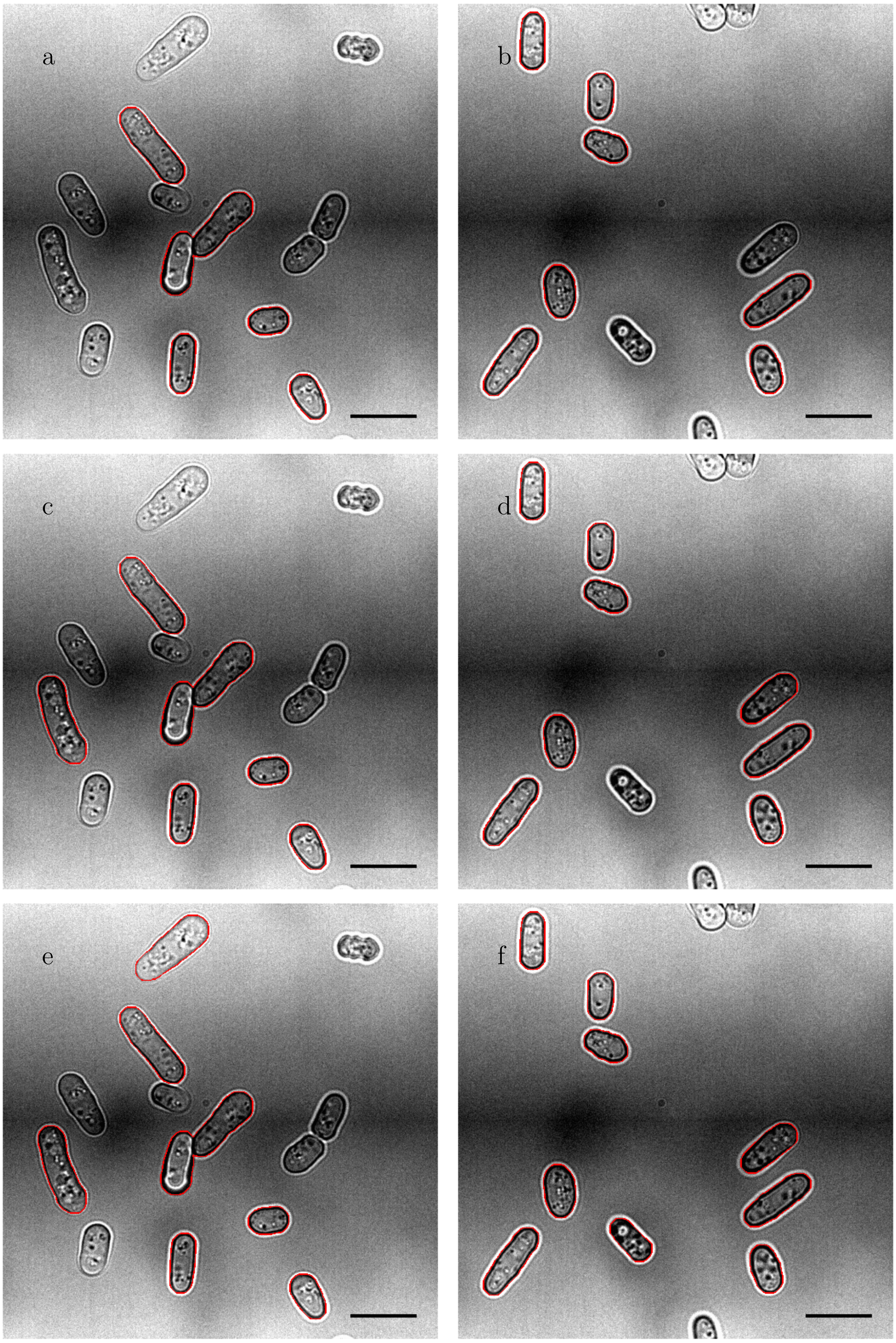


Figure 4.13: Example output image for different first dilation radii: (a)-(b) radius = 0, (c)-(d) radius = 1, (e)-(f) radius = 2. Radii given in pixels. Proposed segmentations shown in red. Scale bars represent 10 µm.

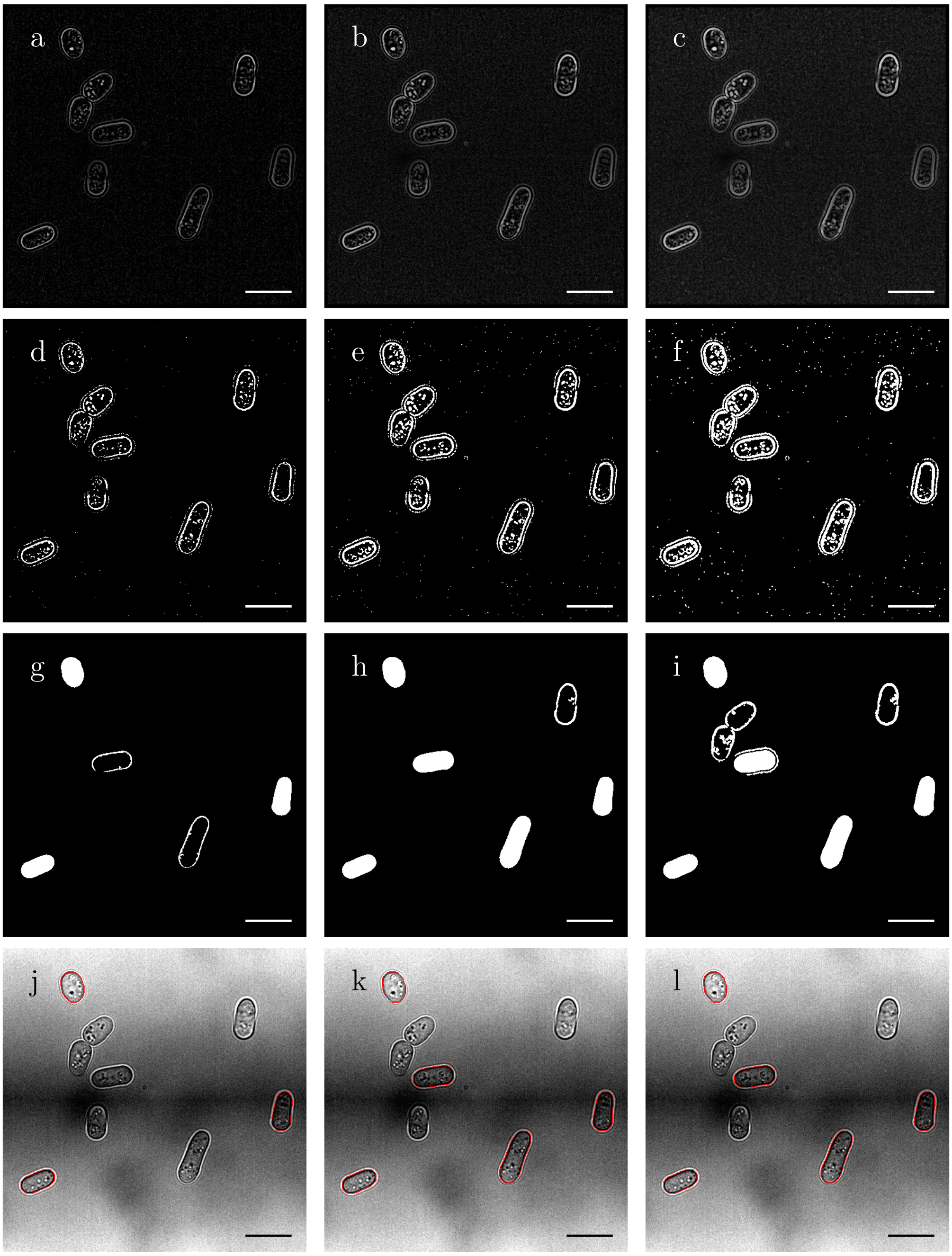


Figure 4.14: Example work-through for different first dilation radii: (L)-(R) radius = 0, radius = 1, radius = 2. Radii given in pixels. (a)-(c) output image of step 3 of figure 4.5; (d)-(f) output image of step 4 of figure 4.5; (g)-(i) output image of step 2 of figure 4.6; (j)-(l) output image of step 7 of figure 4.6. Proposed segmentations shown in red. Scale bars represent 10 µm.

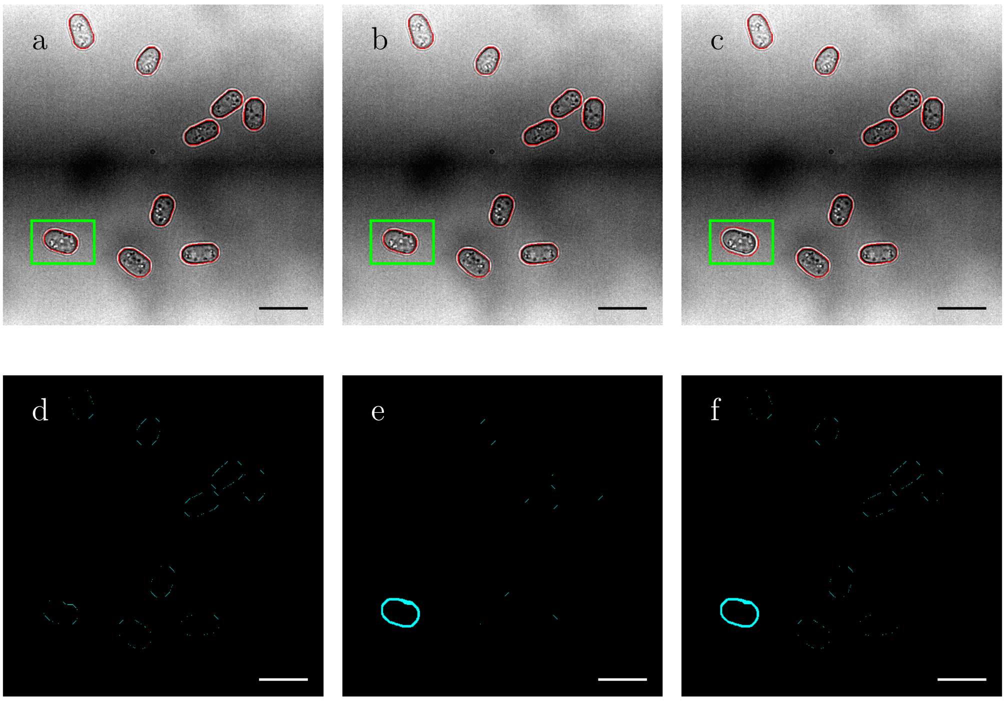


Figure 4.15: Example output of first dilation radius. (a)-(c) output images when the radius of the first dilation is (a) 0, (b) 1, and (c) 2. (d)-(f) differences between proposed segmentation masks shown in blue. Differences between radii of (d) 0 and 1, (e) 0 and 2, (f) 1 and 2. Scale bars represent 10 µm.

to note that when J is 0.95, less than a fifth of the proposed segmentations are classed as true positive segmentations for radii of 0-2 (table 4.5). Less than a fifth of the proposed segmentations are deemed to be false negatives when J is 0.90 and this false to less than a tenth when J is 0.85, or less than a twentieth for radii of 0-1 (table 4.5).

Taking everything into account, a radius of 1 is chosen for the first dilation. It balances the need of a dilation to promote the completion of potential segmentations' perimeter, ready for the filling of regions, with not over-dilating and connecting potential regions which would be excluded during area-based filtering.

4.3.2.5 Disk radius for the erosion of potential segmentations

The radius used for the erosion (figure 4.7) must be large enough to separate touching cells, but small enough to not erode small cells enough to disappear. This is why large intervals were first assessed before 'fine-tuning' of the radius was attempted. The $F_{0.5}$ scores are shown in table 4.6. The minimum area-based filters are set to 0 to remove nothing, while the maximum area-based filter is set to 1,050,000, larger than size of the entire frame $(1,024 \cdot 1,024 = 1,048,576)$, so no segments are deemed too large and removed.

The first result to assess is the effect that removing the area-based filtering has on the segmentation algorithm. Table 4.5 (first dilation radius of 1) and table 4.6 (erosion radius of 20) have the same value for all parameters except for the area-based filters which have been removed in table 4.6. The data shows that the area-based filters do not remove any potential segmentations; however, it highlights the need to determine which false positive proposed segmentations could theoretically be removed by optimising the area-based filters. This means that visual analysis is also necessary.

There is little difference in the number of true positive segmentations while the erosion radius is between 0 and 25, especially at lower J values (table 4.6). This suggests that the erosion is not responsible for separating touching cells; at least not while the erosion radius is less than 30. Visual analysis confirms that a radius of 20 does not separate cells, but radii of 25 and 30 do (figures 4.16a-c). It also highlights that the post-erosion dilations cause the proposed segmentations to be highly inaccurate at the point of connection (figures 4.16d-f). Even though the erosion and dilation steps cannot be used to meaningfully separate touching cells, they can be used to remove inaccurate proposed segmentations. This is where Ver3.2 is used.

						J value					
Erosion		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
radius	**PS**	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$
0	103584	668	0.0080	659	0.0079	649	0.0078	557	0.0067	136	0.0016
5	1762	668	0.4209	662	0.4171	649	0.4089	496	0.3125	112	0.0706
10	685	669	0.9223	665	0.9167	658	0.9071	608	0.8382	149	0.2054
15	685	672	0.9264	665	0.9167	654	0.9016	524	0.7224	118	0.1627
20	685	672	0.9264	665	0.9167	655	0.9030	559	0.7706	102	0.1406
25	701	667	0.9035	661	0.8954	650	0.8805	553	0.7491	65	0.0881
30	825	208	0.2484	167	0.1994	124	0.1481	67	0.0800	1	0.0012
35	68	8	0.0345	5	0.0216	2	0.0086	0	0.0000	0	0.0000
6	858	668	0.7733	662	0.7664	650	0.7525	500	0.5788	113	0.1308
7	708	669	0.8994	665	0.8941	657	0.8833	606	0.8147	148	0.1990
8	695	668	0.9108	662	0.9026	650	0.8863	502	0.6845	115	0.1568
9	691	669	0.9162	662	0.9066	650	0.8902	505	0.6916	114	0.1561
10	685	669	0.9223	665	0.9167	658	0.9071	608	0.8382	149	0.2054
11	684	669	0.9233	662	0.9136	652	0.8998	509	0.7025	116	0.1601
12	683	669	0.9243	662	0.9146	652	0.9008	514	0.7101	116	0.1603
13	683	669	0.9243	662	0.9146	654	0.9036	518	0.7157	115	0.1589
14	685	672	0.9264	667	0.9195	660	0.9098	616	0.8492	150	0.2068
15	685	672	0.9264	665	0.9167	654	0.9016	524	0.7224	118	0.1627
16	685	672	0.9264	666	0.9181	655	0.9030	536	0.7389	117	0.1613
17	686	672	0.9254	668	0.9199	661	0.9102	617	0.8496	143	0.1969
18	686	672	0.9254	665	0.9157	655	0.9020	546	0.7519	112	0.1542
19	685	672	0.9264	665	0.9167	656	0.9043	551	0.7596	108	0.1489
20	685	672	0.9264	665	0.9167	655	0.9030	559	0.7706	102	0.1406
21	686	671	0.9240	668	0.9199	663	0.9130	626	0.8620	124	0.1708
22	688	673	0.9247	667	0.9165	656	0.9013	570	0.7832	99	0.1360
23	690	671	0.9199	665	0.9117	657	0.9007	571	0.7828	85	0.1165
24	698	670	0.9106	666	0.9051	657	0.8929	609	0.8277	88	0.1196

Table 4.6: Table of $F_{0.5}$ scores and number of true positive (tp) segmentations for the optimisation of the erosion parameter over a range of J values. The number of proposed segmentations (PS) is also given. Radii given in pixels.

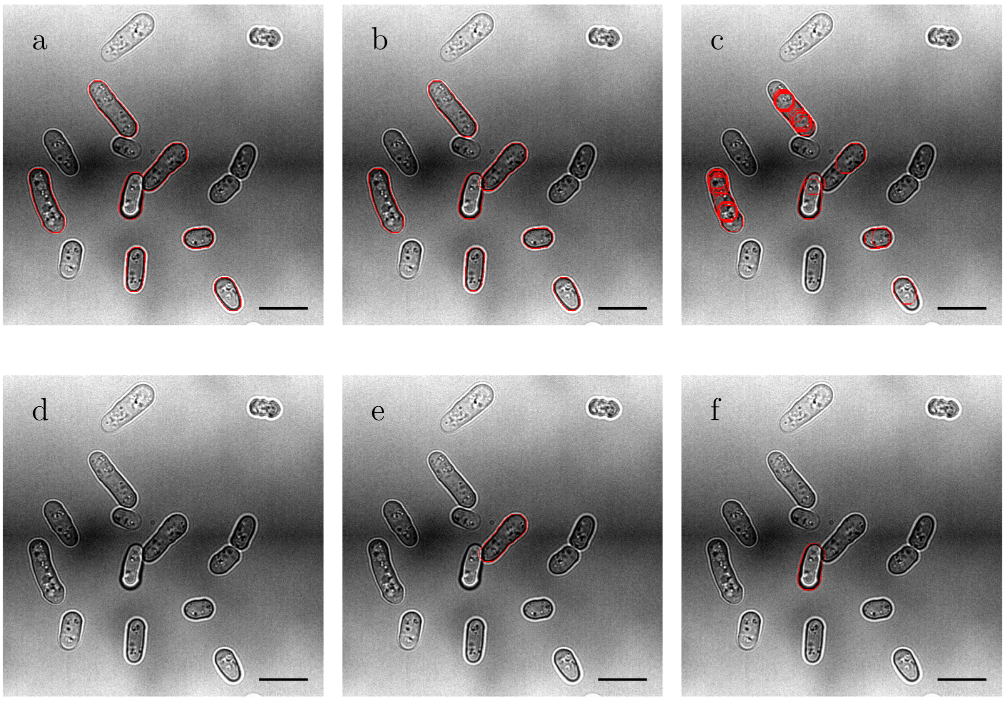


Figure 4.16: Example output of erosion radius exploration. (a)-(c) erosion radius of (a) 20, (b) 25, and (c) 30. (d)-(f) singular segmentations of (b): (d) input image, (e) first proposed segmentation, (f) second proposed segmentation. Radii given in pixels. Proposed segmentations shown in red. Scale bars represent 10 µm.

Variation for final output image

As the erosion step is not used to separate touching cells, the radius for the output proposed segmentations can be selected based on the number of true positive segmentations and visual analysis. This method is chosen because the $F_{0.5}$ score reflects the lack of area-based filters. The larger interval study shows that radii 15 and 20 result in the largest number of true positive proposed segmentations when J is 0.75, and a radius of 10 performs as well as 15 and 20 when J is 0.80 (table 4.6). For higher J values, a radius of 10 results in more true positives than radii of 15 or 20 (table 4.6). Further investigations were carried out on radii covered by the range of 6-24 (results shown in table 4.6).

Examination of the data in table 4.6 and figure 4.17 shows that there is minimal difference in the number of true positive proposed segmentations while J $\leq$ 0.85. When J is 0.90, radii of 7, 10, 14, 17, 21, and 24 all cause definite peaks;

this trend is also seen when J is 0.95 (figure 4.17). The peaks, when J is 0.95. are lower for radii of 21 and 24 which eliminates them as the ideal value. Radii of 7 and 10 perform better than radii of 21 and 24 when J is 0.95; however, they perform worse when J is 0.90 (compared to a radius of 21; table 4.6).

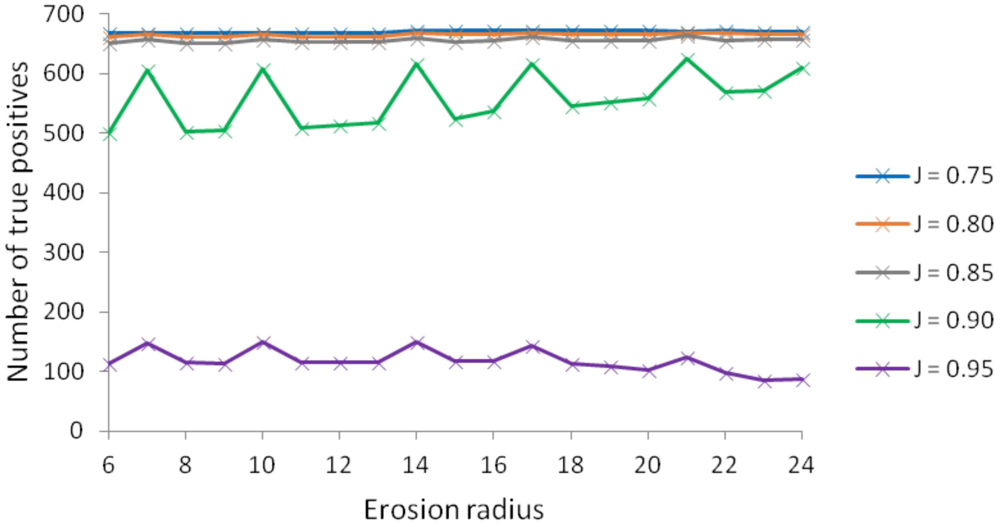


Figure 4.17: Graph of the number of true positive segmentations for the optimisation of the erosion radius parameter over a range of J values. Radii given in pixels.

A radius of either 14 or 17 are the most suitable candidates for this erosion. Only one proposed segmentation differs enough between the two radii to be considered a true positive when the radius is 17 but not when the radius is 14 while J is 0.80. This proposed segmentation is found in the image shown in figure 4.18. Visually, there is no discernible difference between figures 4.18a and 4.18b. The fact that a radius of 14 produces a better $F_{0.5}$ score when J is 0.75 or 0.95 and is only ever missing one proposed segmentation compared to a radius of 17 when J is between 0.80 and 0.90 means that a radius of 14 was selected over 17.

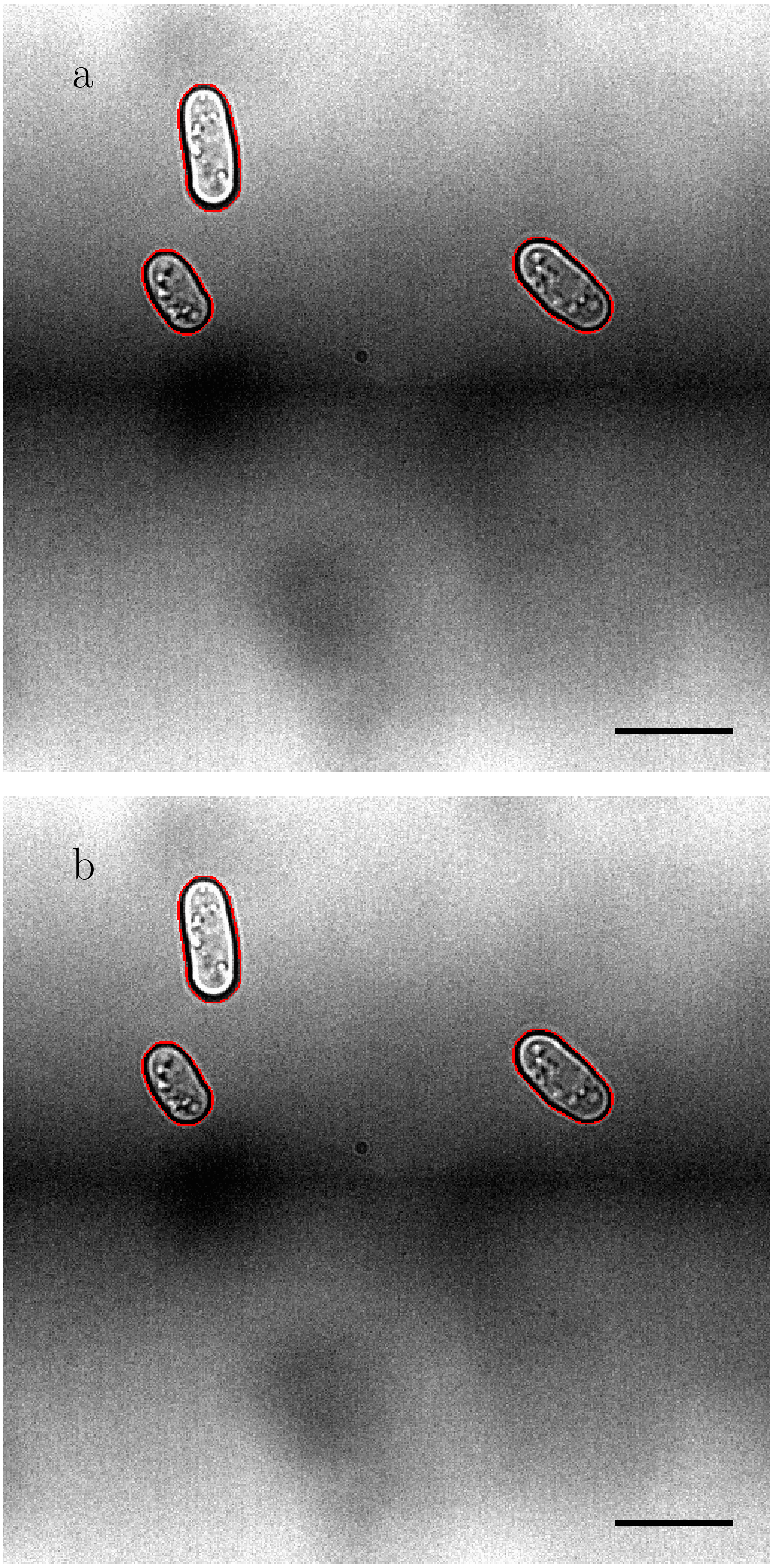


Figure 4.18: Output image when erosion radius is (a) 14, and (b) 17. Radii given in pixels. Proposed segmentations in red. Scale bars represent 10 µm.

Variation for removing touching cells

Touching cells can be identified based on the size of potential segmentations once eroded, providing the radius of the disk used for said erosion does not cause breaks in typical, singular cells (as seen in figure 4.16c). Potential segmentations that become two segments post-erosion can be excluded as they are likely to be touching cells, provided that the disk diameter does not cause over-erosion. Cells undergoing cytokinesis will be either segmented as two cells based on the steps proceeding the erosion if they are already divided; however, if only one proposed segmentation is produced, the additional erosion will determine if the cells have separated and ought to be considered as two cells or if the cell is still one cell.

Based on visual analysis, a radius of 20 separates some touching cells while a radius of 25 leads to several individual cells becoming multiple proposed segments. This means that a radius of 25 cannot be used to identify touching cells as it is too large, but a radius around 20 is suitable. As a radius of 21 peaks in the total number of true positive segmentations at higher J values, it was selected as the secondary erosion disk radius.

4.3.2.6 Region area sizes for the post-erosion area-based filters

The first area-based filters to be optimised are those of the potential segmentations after the erosion (figure 4.7). With the pre-erosion filters kept at 0 and 1,050,000, the $F_{0.5}$ scores are a suitable measure for the success of the algorithm. The results are displayed in tables 4.7 and 4.8.

Maximum area filter

The maximum area-based filter was explored before the minimum area was set. The minimum area filter was set as 0 and the results area shown in table 4.7.

An area of 20,000 is the smallest value at which the $F_{0.5}$ score is maximal for the large interval study excluding when J is 0.95 (table 4.7). When J is 0.95,

Maximum		**J value**				
area (in 000s)	**PS**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
0	0	0.0000	0.0000	0.0000	0.0000	0.0000
10	681	0.9277	0.9222	0.9125	0.8516	0.2077
20	682	0.9281	0.9225	0.9129	0.8520	0.2075
30	682	0.9281	0.9225	0.9129	0.8520	0.2075
40	682	0.9281	0.9225	0.9129	0.8520	0.2075
11	681	0.9277	0.9222	0.9125	0.8516	0.2077
12	682	0.9281	0.9225	0.9129	0.8520	0.2075
13	682	0.9281	0.9225	0.9129	0.8520	0.2075
14	682	0.9281	0.9225	0.9129	0.8520	0.2075
15	682	0.9281	0.9225	0.9129	0.8520	0.2075
16	682	0.9281	0.9225	0.9129	0.8520	0.2075
17	682	0.9281	0.9225	0.9129	0.8520	0.2075
18	682	0.9281	0.9225	0.9129	0.8520	0.2075
19	682	0.9281	0.9225	0.9129	0.8520	0.2075

Table 4.7: Table of $F_{0.5}$ scores for the optimisation of the maximum area-based filter post-erosion over a range of J values. The number of proposed segmentations (PS) is also given. Areas given in 000s of pixels.

an area of 10,000 provides a higher $F_{0.5}$ score (0.2077 compared to 0.2075; table 4.7) as the proposed segmentation included when the area is increased to 20,000 pixels has a J value between 0.90 and 0.95. Examining smaller intervals shows that an area of 12,000 permits the largest true positive proposed segmentation to be included (table 4.7). It is also relevant to note that even with an area of 40,000 for the maximum area filter, the algorithm's success does not change. A maximum area of 12,000 pixels (relative to the 100x-oil-Z dataset) was chosen for the post-erosion area-based filter.

Minimum area filter

With the maximum area filter set at 12,000, the minimum area was explored. The results are shown in table 4.8.

Table 4.8 shows that increasing the minimum area filter from 0 to 1,000 removes only one potential segmentation. To ensure that the decrease in the

Minimum		J value				
area (in 00s)	**PS**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
0	682	0.9281	0.9225	0.9129	0.8520	0.2075
10	681	0.9291	0.9236	0.9139	0.8529	0.2077
20	641	0.9142	0.9084	0.8997	0.8447	0.2101
30	392	0.7821	0.7760	0.7637	0.7230	0.2057
40	166	0.5190	0.5126	0.5061	0.4771	0.1612
1	682	0.9281	0.9225	0.9129	0.8520	0.2075
2	682	0.9281	0.9225	0.9129	0.8520	0.2075
3	682	0.9281	0.9225	0.9129	0.8520	0.2075
4	682	0.9281	0.9225	0.9129	0.8520	0.2075
5	682	0.9281	0.9225	0.9129	0.8520	0.2075
6	682	0.9281	0.9225	0.9129	0.8520	0.2075
7	682	0.9281	0.9225	0.9129	0.8520	0.2075
8	682	0.9281	0.9225	0.9129	0.8520	0.2075
9	681	0.9291	0.9236	0.9139	0.8529	0.2077
10	681	0.9291	0.9236	0.9139	0.8529	0.2077
11	681	0.9291	0.9236	0.9139	0.8529	0.2077
12	681	0.9291	0.9236	0.9139	0.8529	0.2077
13	680	0.9287	0.9232	0.9135	0.8525	0.2079
14	679	0.9284	0.9228	0.9131	0.8251	0.2082
15	677	0.9277	0.9221	0.9138	0.8540	0.2086
16	672	0.9259	0.9203	0.9119	0.8531	0.2098
17	666	0.9237	0.9181	0.9096	0.8533	0.2098
18	660	0.9215	0.9158	0.9073	0.8520	0.2098
19	653	0.9188	0.9131	0.9045	0.8488	0.2115

Table 4.8: Table of $F_{0.5}$ scores for the optimisation of the minimum area-based filter post-erosion over a range of J values. The number of proposed segmentations (PS) is also given. Areas given in 00s of pixels.

$F_{0.5}$ scores between 1,000 and 2,000 pixels is not just due to the loss of 40 proposed segmentations, intervals of 100 were examined above and below 1,000.

The loss of another potential segmentation when the area is 1,300 results in a decrease in the $F_{0.5}$ score (table 4.8). As a minimum area of 1,200 pixels (relative to the 100x-oil-Z dataset) is the largest area size with the highest $F_{0.5}$ score while J is less than 0.95, it was chosen for the post-erosion area-based filter.

4.3.2.7 Region area sizes for the pre-erosion area-based filters

The post-erosion area-based filters were set as 1,200 and 12,000 pixels (for the 100x-oil-Z dataset) for the exploration of the pre-erosion area-based filters. The results for the pre-erosion filters are are displayed in tables 4.9 and 4.10.

Maximum area filter

The maximum area-based filter was explored before the minimum area was set. The minimum area filter was set as 0 and the results area shown in table 4.9.

Maximum		**J value**				
area (in 000s)	**PS**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
0	0	0.0000	0.0000	0.0000	0.0000	0.0000
10	576	0.8916	0.8884	0.8806	0.8226	0.1833
20	679	0.9284	0.9228	0.9145	0.8535	0.2082
30	680	0.9287	0.9232	0.9149	0.8539	0.2079
40	681	0.9291	0.9236	0.9139	0.8529	0.2077
50	681	0.9291	0.9236	0.9139	0.8529	0.2077
60	681	0.9291	0.9236	0.9139	0.8529	0.2077

Table 4.9: Table of $F_{0.5}$ scores for the optimisation of the maximum area-based filter pre-erosion over a range of J values. The number of proposed segmentations (PS) is also given. Areas given in 000s of pixels.

The region size for the maximum pre-erosion area-based filter cannot produce a higher $F_{0.5}$ score consistently over the range of J values than those found in table 4.8 (compared with table 4.9). This suggests that this is a redundant part of the algorithm and can be kept as 1,050,000 or removed from the function without affecting performance. To reduce computational effort, the maximum post-erosion area-based filter was removed.

Minimum area filter

With the maximum area filter removed, the minimum area was explored. The results are shown in table 4.10.

Minimum		**J value**				
area (in 00s)	**PS**	**0.75**	**0.80**	**0.85**	**0.90**	**0.95**
0	681	0.9291	0.9236	0.9139	0.8529	0.2077
10	681	0.9291	0.9236	0.9139	0.8529	0.2077
20	681	0.9291	0.9236	0.9139	0.8529	0.2077
30	681	0.9291	0.9236	0.9139	0.8529	0.2077
40	681	0.9291	0.9236	0.9139	0.8529	0.2077
50	657	0.9203	0.9147	0.9061	0.8492	0.2105
60	567	0.8843	0.8780	0.8685	0.8130	0.2060
70	413	0.7956	0.7897	0.7779	0.7306	0.1969

Table 4.10: Table of $F_{0.5}$ scores for the optimisation of the minimum area-based filter pre-erosion over a range of J values. The number of proposed segmentations (PS) is also given. Areas given in 00s of pixels.

The region size for the minimum pre-erosion area-based filter fails to produce a higher $F_{0.5}$ score consistently over the range of J values than those found in tables 4.8 and 4.9 (compared with table 4.10). This suggests that this is also a redundant part of the algorithm and can be kept as 0 or removed from the function without affecting performance. To reduce computational effort, the minimum post-erosion area-based filter was removed.

4.3.3 Limitations of the novel segmentation algorithm based on imaging conditions

With a fully optimised algorithm for the 100x-oil-Z dataset, the next step was to explore the affect of the imaging conditions on the success of the segmentation algorithm. The $F_{0.5}$ scores for J values of 0.75-0.95, in intervals of 0.05, can be found in table 4.11. Figure 4.19 shows example segmentations for each dataset.

The largest $F_{0.5}$ scores for all J values occur for the 100x-oil-Z dataset (table 4.11) which is to be expected as the algorithm was optimised for the dataset. None of the other datasets achieve an $F_{0.5}$ above 0.900, with the 60x-oil-Z dataset not even managing to reach the 0.8000 (table 4.11). This reflects the trend of the mean and median J values produced by the human participation study in chapter

							J value					
	Number		**0.75**		**0.80**		**0.85**		**0.90**		**0.95**	
Dataset	**of cells**	**PS**	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$	**tp**	$F_{0.5}$
100x-oil-Q	402	220	206	0.8034	203	0.7917	200	0.7800	175	0.6825	18	0.0702
100x-oil-Z	888	681	671	0.9291	667	0.9236	660	0.9139	616	0.8529	150	0.2077
60x-oil-Z	391	188	155	0.6780	115	0.5031	48	0.2100	5	0.0219	0	0.0000
60x-air-Z	203	121	112	0.8151	81	0.5895	38	0.2766	6	0.0437	0	0.0000

Table 4.11: Table of $F_{0.5}$ scores and number of true positive (tp) segmentations for the each of the datasets over a range of J values. The number of cells and proposed segmentations (PS) are also given.

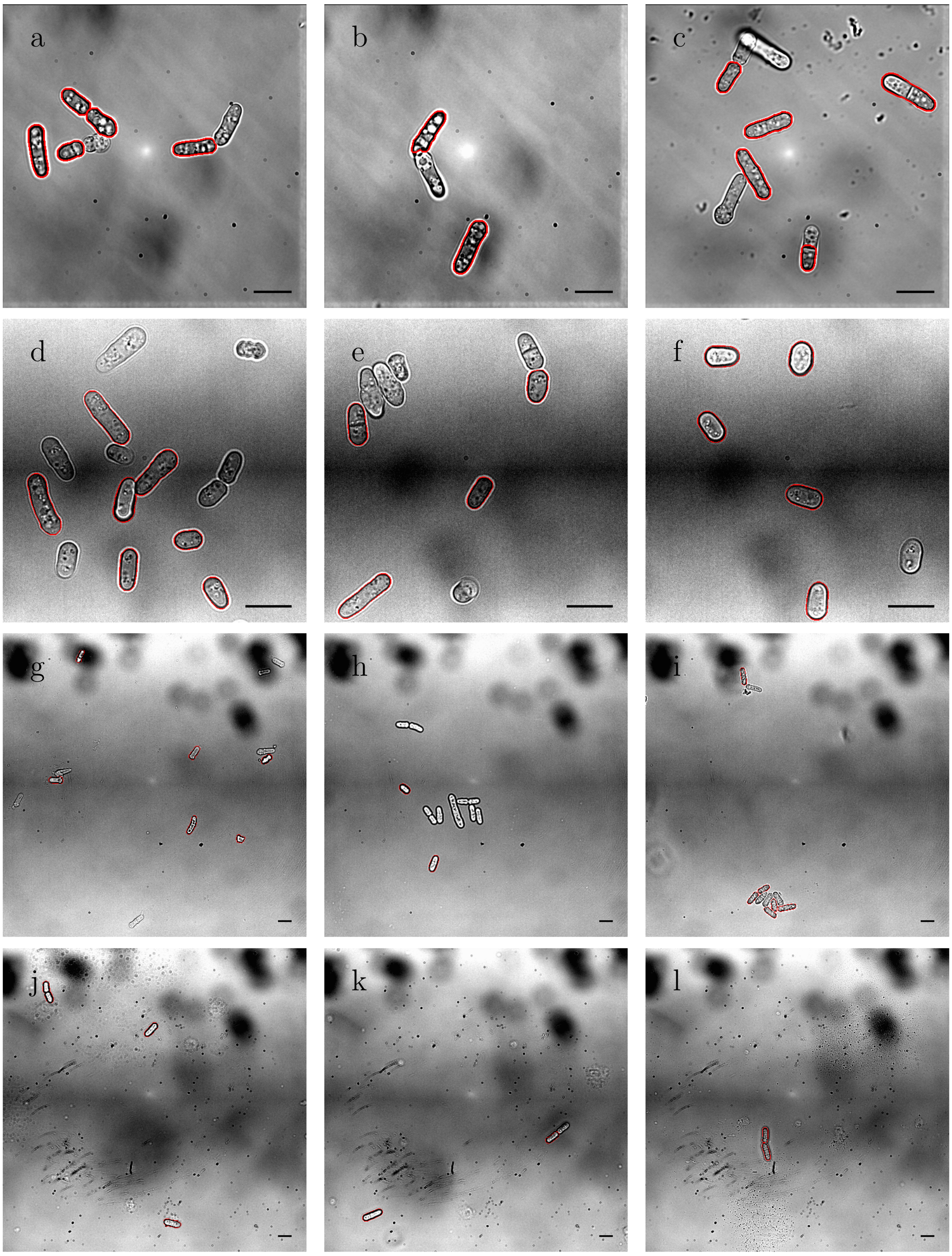


Figure 4.19: Example output images for each dataset. (a)-(c) 100x-oil-Q dataset, (d)-(f) 100x-oil-Z dataset, (g)-(i) 60x-oil-Z dataset, (j)-(l) 60x-air-Z dataset. Proposed segmentations in red. Scale bars represent 10 µm.

3.3.5 where the 60x-oil-Z subset showed a mean and median of 0.69 for manual segmentation (table 3.1).

Both of the datasets with 100x magnification are segmented better than the 60x magnification datasets even though the camera resolution differs between the 100x magnification datasets. The difference in camera resolution, when combined with the magnification, means that the 100x magnification datasets have the highest and lowest level of detail which does not support the hypothesis that the magnification or resolution affects the novel algorithm's segmentation success.

As the 60x-air-Z dataset produces larger $F_{0.5}$ scores than the 60x-oil-Z dataset (table 4.11), it is possible that the novel segmentation algorithm performs better on datasets with air as the immersion medium rather than oil. This reflects the manual segmentation results in table 3.1 and is interesting as oil as an immersion medium produces a clearer image as the refraction of light is reduced [39].

Visual analysis (examples given in figure 4.19) shows that only the 60x-oil-Z dataset has several false positive segmentations that do not correspond to a cell (figure 4.19g), as well as those which are associated with cells (such as those in figures 4.19b-d and 4.19j). This could be due to the growth or mounting of the cells, marring the dataset and influencing the effectiveness of the segmentation algorithm. All datasets show at least one example of touching cells being correctly segmented, even if not all of the cells involved were kept.

Variations in focus across datasets also influences the $F_{0.5}$ scores. The focus determines the size of the halo around the cells and this affects the novel segmentation algorithm's success. The ground truth segmentations are based on the edge of the cell wall; typically with a definitive white halo. Sometimes, such as in figure 4.19f and the 60x magnification datasets (excluding figure 4.19i), the focus results in a narrow halo. In figure 4.19f, the effect is to reduce the J values for the proposed segmentations, in turn lowering the $F_{0.5}$ score. For the datasets with 60x magnification, it appears to reduce the number of proposed segmentations as well as affect the J values. This could also be why the manual segmentations disagree in these datasets so much in chapter 3.3.5.

Overall, the total number of proposed segmentations for each dataset is approximately half of the total number of cells present for all datasets except for the 100x-oil-Z dataset where it is roughly three-quarters (table 4.11). Of the proposed segmentations, most are deemed to be true positives when J is 0.75. The 100x magnification datasets maintain a high number of true positive classified segmentations even when J is 0.85, with a slightly sharper decrease when J is 0.90. True positive classifications fall more dramatically for the datasets with 60x magnification. This could be a result of the 60x magnification datasets being transmitted light microscopy images rather than phase-contrast microscopy images, like the datasets with 100x magnification. It is likely that the increased contrast of phase-contrast microscopy is beneficial for the novel segmentation algorithm; however, the imaging conditions have little impact on its success providing the datasets are fully focussed to imitation the 100x magnification datasets or figures 4.19i and 4.19l.

It is not surprising that the imaging conditions used here did not affect the performance of the segmentation algorithm as using the Rayleigh criterion (equation 4.6) shows that the lenses resolve to <1 µm. This means that the details in the image are resolved to a level below the size of the camera sensors pixel (chapter 2). However, the pixel size is now the limiting factor for the imaging condition, and therefore, a difference between the CMOS and CCD cameras could be expected. Results in table 4.11 do suggest that the pixel size does play a role in the segmentation algorithm's success, but no more than the magnification.

$$r = \frac{1.22\lambda}{2\mathrm{NA}} \tag{4.6}$$

4.3.4 Comparison of the novel segmentation algorithm to PombeX

The novel segmentation algorithm is compared to PombeX, an existing segmentation algorithm for *S. pombe* cells.

4.3.4.1 PombeX segmentation algorithm and their dataset

As the exact analysis method used in [62] and [110] is unknown, the PombeX program was run on the PombeX dataset and analysed as described in chapter 2.5.1. The results are shown in figure 4.20.

A total of 1,423 segmentations were proposed for the 1,422 cells, of which 965 were deemed acceptable (figure 4.20). This produces an $F_{0.5}$ score of 0.6782. On a different dataset, PombeX finds 94.5% of all cells and correctly segments (<10% pixel mismatch compared to the ground truth images) 70.8% of all cells when the fluorescence images are not used [110]. On the released dataset, [73], 67.9% of all cells were segmented to a visually acceptable standard.

4.3.4.2 PombeX segmentation algorithm with the new datasets

The 100x-oil-Z dataset contains 887 cells. The PombeX segmentation algorithm produces 3,027 potential segmentations over the entire dataset with only one of these being deemed a 'visually perfect' segmentation (figure 4.21a) and another 17 potential segmentations being classified as 'acceptable' (figure 4.21b). This means that the $F_{0.5}$ score for PombeX on the 100x-oil-Z dataset is 0.0069, which is significantly lower than that of the novel segmentation algorithm, even with a J value of 0.95 ($F_{0.5} = 0.2077$ when J is 0.95; $F_{0.5} = 0.8529$ when J is 0.90; table 4.11).

Ensuring that the poor segmentation success of PombeX was not based on the imaging condition, the PombeX program was run on the hybrid dataset. The

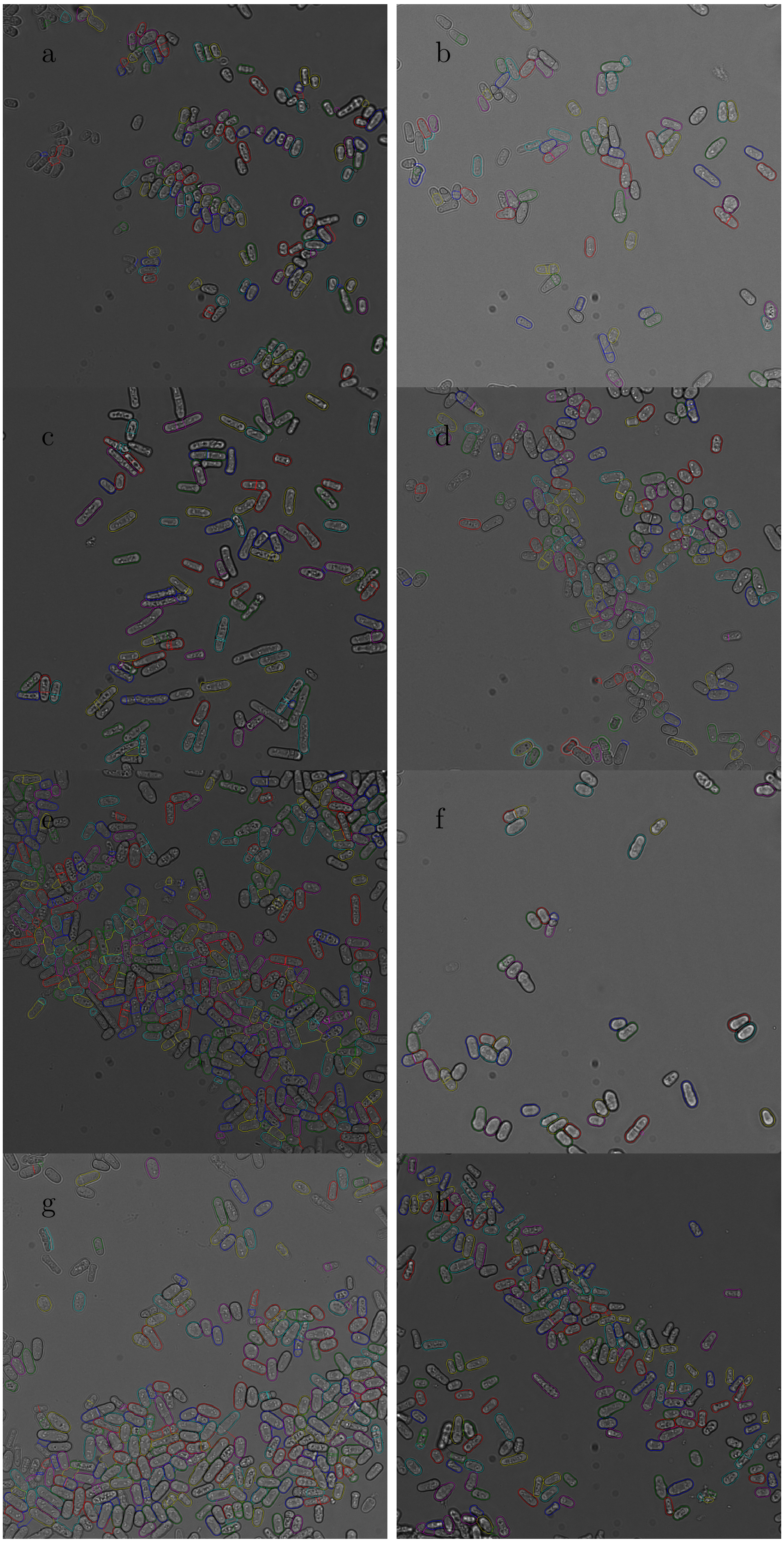


Figure 4.20: Output images of the PombeX program on the PombeX dataset.

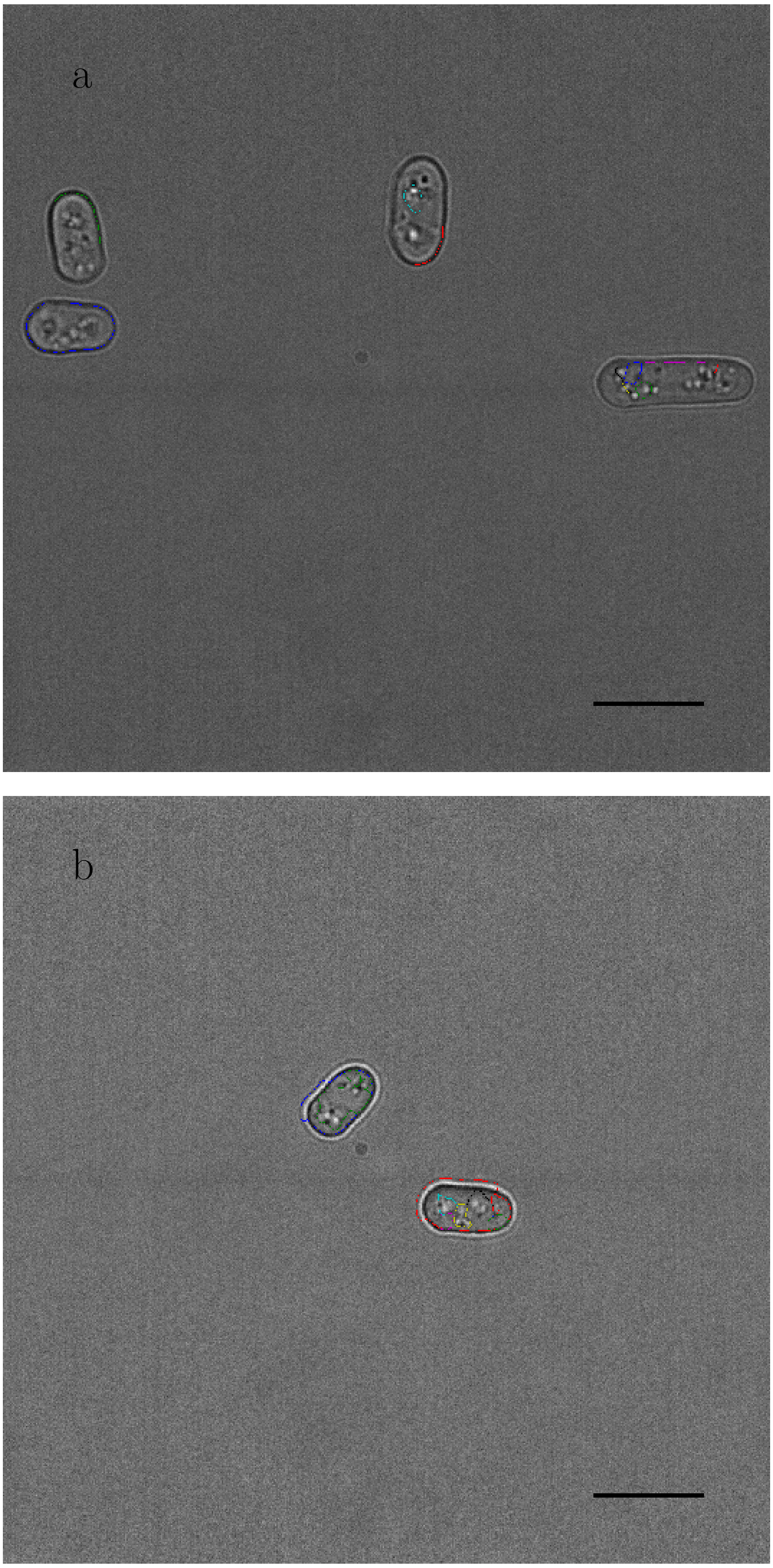


Figure 4.21: Example output images of the PombeX program on the 100x-oil-Z dataset. (a) the only 'perfect' segmentation in dark blue, centre left. Scale bars represent 10 µm.

results are shown in table 4.12 with an example image for each dataset in figure 4.22.

Original dataset	**tp**	**fp**	**fn**	$F_{0.5}$ **score**
100x-oil-Q	2	13	16	0.1282
100x-oil-Z	0	68	36	0.0000
60x-oil-Z	3	607	8	0.0061
60x-air-Z	1	58	13	0.0200

Table 4.12: Table of segmentation success of PombeX on the hybrid dataset. The number of true positives (tp), false positives (fp), and false negatives (fn) given.

Table 4.12 demonstrates that there is no clear correlation between the imaging conditions and the success rate of the segmentation algorithm. This includes the type of microscopy, as the transmitted light microscopy images were segmented with comparable success to the phase contrast microscopy images. The most important data to consider from table 4.12 is the number of false positives. These are the major influences on the $F_{0.5}$ scores.

The 100x-oil-Q dataset does produce the highest $F_{0.5}$ score (table 4.12); however, there is no clear visible reason for this (figure 4.22). This could be coincidental (the hybrid dataset is constructed from tiny subsets of the larger datasets, which could lead to unbalanced results); however, there are two other potential factors. First, the QuantEM CCD camera is only used in this dataset. The camera's resolution, along with the magnification, gives the largest level of detail (in terms of how many pixels make up a micrometre). This links to the second factor; the input image for this dataset is 512 pixels by 512 pixels. This could be a factor as PombeX's output images are all scaled to a maximum size of 1,066 pixels by 1,066 pixels, but the 100x-oil-Q images are only 1,028 pixels by 1,028 pixels. Not exceeding the maximum output image size, and almost doubling the image, could be the cause.

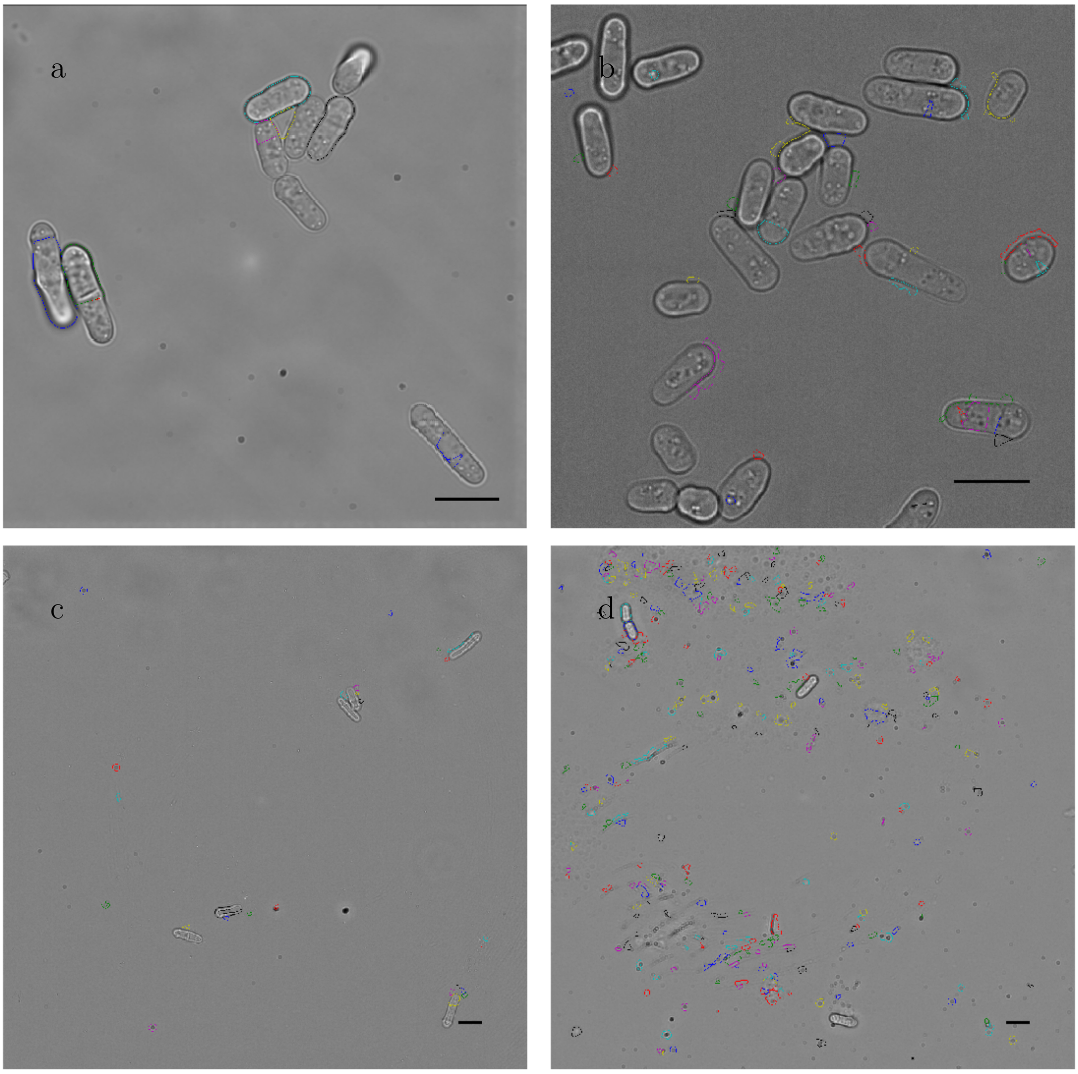


Figure 4.22: Example output images of the PombeX program on the hybrid dataset. Images from: (a) 100x-oil-Q, (b) 100x-oil-Z, (c) 60x-oil-Z, and (d) 60x-air-Z. Scale bars represent 10 µm.

4.3.4.3 Novel segmentation algorithm and PombeX's dataset

The optimised novel algorithm was run on the PombeX dataset with the magnification and resolution values found in the hybrid dataset. An estimated magnification of 100x and resolution of 16 µm per pixel produced the best proposed segmentations which are shown in figure 4.23.

A total of 204 segmentations were proposed for the 1,422 cells, of which 165 were deemed acceptable (figure 4.23). This produces an $F_{0.5}$ score of 0.3686. This is less than half that which the PombeX program achieves ($F_{0.5} = 0.6782$);

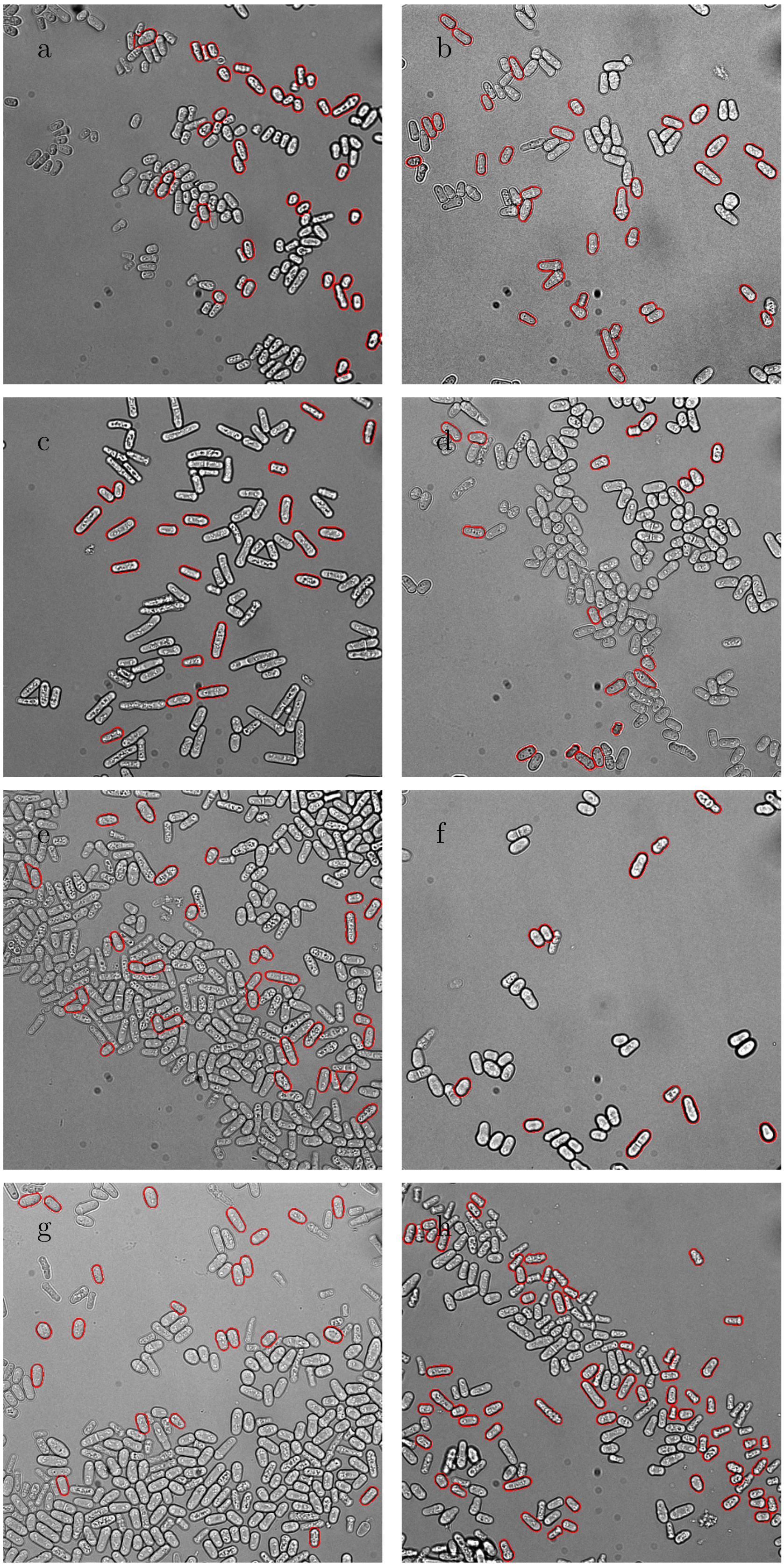


Figure 4.23: Output images of the Outline program on the PombeX dataset when the magnification is set as 100x and resolution as 16 µm per pixel.

however, the magnification and resolution are assumed. Using the correct values may increase the novel algorithm's performance, but it is highly unlikely to reach that of the PombeX program.

Unlike the PombeX based analysis of the new datasets, the low $F_{0.5}$ score is due to a low number of proposed segmentations, instead of a large number of false positives. The majority of the false positive segmentations found in figure 4.23 are from incorrectly segmented touching cells. However; there are no false positive segmentations that do not correspond to cells.

4.3.4.4 Conclusions of comparing PombeX and the novel segmentation algorithm

PombeX definitely performs well on its own dataset, as expected; however, it is not usable on the datasets provided in this thesis. Not only does it not segment enough cells, it also generates an exorbitant number of false positives that would require a researcher to manual remove them from the output before further analysis can be completed.

In comparison, the novel segmentation algorithm described here does not perform as well on other datasets as it does on the one it was developed on, but produces far fewer false positives. There is the potential for altering the parameters to reduce the occurrence of false positives too for each dataset if the researcher decides the proposed segmentations do not meet their criteria. The number of useful proposed segmentations compared to all proposed segmentations is substantially higher than PombeX (81% compared to 1% respectively). This is preferable to manual analysis as taking more images is less time consuming, even including the time required for the removal of the 20% false positive segmentations.

The output images from the algorithms is also important to note. PombeX uses seven different colours to outline their segmentations, including black. It is not possible for researchers to identify and critique these segmentations. The novel segmentation algorithm uses one colour for all of the segmentations, and can be changed from red to any other desired colour by the end-user. Unlike PombeX, the output images are not scaled by the novel segmentation algorithm which allows for easy modification (such as adding scale bars for publications).

4.4 Conclusions

The aims and objectives of this chapter were:

- To develop a novel segmentation algorithm (which will be called Outline) primarily for the segmentation of *S. pombe* cells from phase contrast microscopy images.
- For Outline to perform better than PombeX, the available segmentation algorithm specifically designed for segmentation of *S. pombe* cells from transmitted light microscopy.
- To define any clear limitations of the novel segmentation algorithm based on imaging conditions.

All of these targets were achieved. The novel segmentation algorithm does segment *S. pombe* cells from phase contrast microscopy images as well as from transmitted light microscopy images. No clear limitations were uncovered after exploring: magnification, image resolution, type of camera chip, and immersion media.

Whether the Outline algorithm performs better than the PombeX program is less definitive. On the PombeX dataset, the PombeX program performs over twice as well as the Outline algorithm; however, on the 100x-oil-Z and hybrid datasets, the Outling algorithm drastically outperforms the PombeX program. As the PombeX program produces a large number of false positive segmentations, it is less helpful for the end-user as they have to check all the analysis. The Outline algorithm produces very few false positives which means that the end-user needs to provide more input images which is less time consuming.

CHAPTER 5

Measuring Cell Lengths and Widths

5.1 Introduction

Cell geometry is often a desired metric from microbiological experiments. It can provide a simple manner in which to identify if a mutation or stressor affects cell growth (especially for *S. pombe* as described in chapter 1). The geometric properties of segments provided by algorithms such as the one described in chapter 4 can be measured more robustly by an automated algorithm than manually. This is important as there are several variables in how cells are measured, for example, how curved/bent cells are treated, or where width measurements are taken. Repeatability would also be improved by a fully automated measurement method; therefore, data could be shared and compared between researchers and laboratories with confidence. For this to be true, a method which is accurate in making measurements relating to segment length and width is required.

5.1.1 Defining geometric properties

Three of the most commonly desired measurements used in cell biology are area, length, and width. While the area of a segment is easy to calculate by counting the number of pixels within it, measuring the length and width of a segment prove more complex. It is also unhelpful as it is common for a methodology to just state that a length or width measurement was taken, without giving details as to how [111, 112].

The problem with taking length and width measurements is in their definitions. Length is generally thought of as the largest distance across a region and the width as the distance from one side to the other (taking the length as the centre). These definitions are easy to use when considering a regular polygon; however, once irregular shapes, especially ones with curves, are considered, it complicates matters.

The width is often measured perpendicular to the length, but is sometimes considered the narrowest part of the segment (providing it intersects the length line; figure 5.1). It also tends to be measured as a straight line. The length can be considered as the longest straight line spanning the segment regardless of shape (figure 5.2a-c), or it could be the longest continuous line that follows the shape of the segment by remaining central (figure 5.2d-f).

5.1.2 Some currently used methods for measuring lengths and widths

There are several algorithms that are designed to measure the length and width of segments.

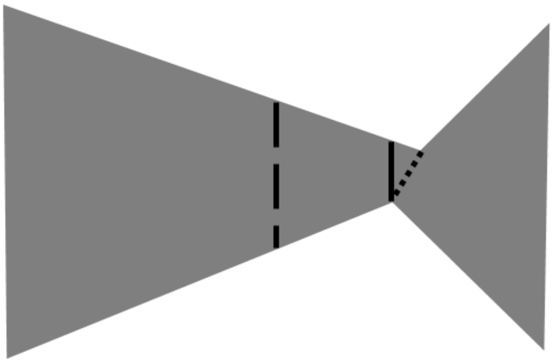


Figure 5.1: Examples of where width measurements can be made. The perpendicular line halfway along the length line (dashed line), the narrowest option while remaining perpendicular to the length line (solid line), and the narrowest option (dotted line).

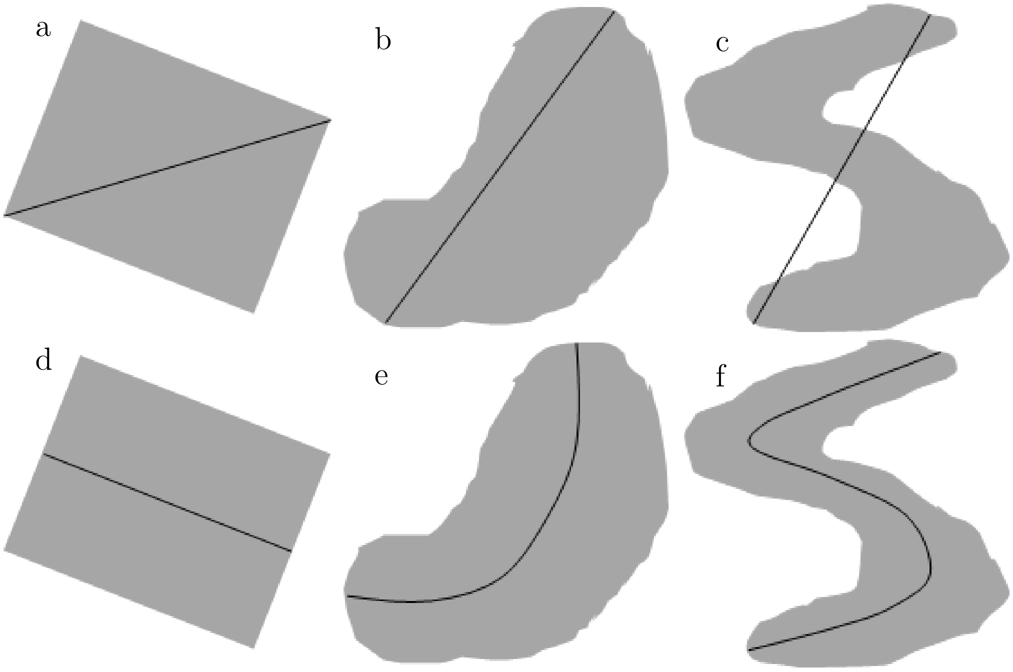


Figure 5.2: Examples of how length measurements can be made. Length measurements shown in black. (a)-(c) straight line length measurements; (d)-(f) length taken along segment centre.

5.1.2.1 Counting pixels of the medial axis

Finding the medial axis via skeletalisation and then counting the number of pixels is a rarely used method for determining the length of a segment [113]. There are two problems with this method. First, there is no accompanying method to make a width measurement. Second, their are often problems in the skeletalisation of the segment. Spurs may appear due to irregularities in the segment shape and the skeleton never fully reaches the ends of the segment (figure 5.3). These things all affect how many pixels are present.

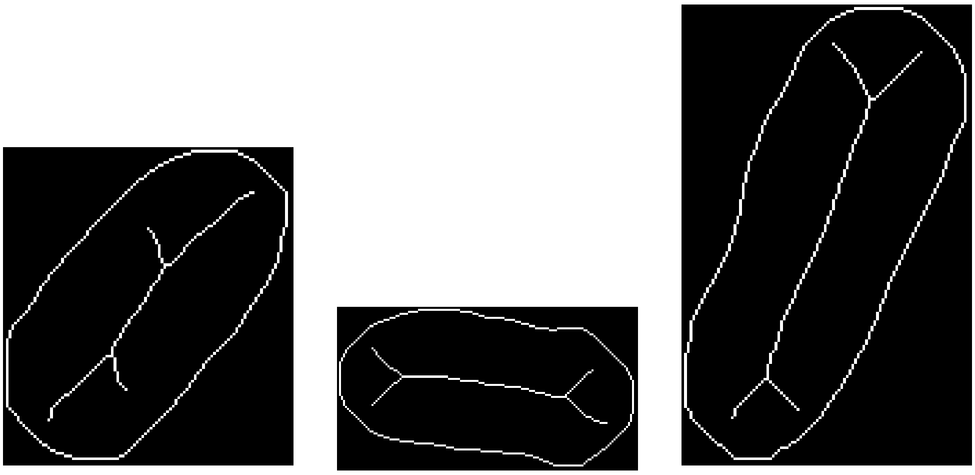


Figure 5.3: Diagram of skeletalised segments within their segment perimeter.

5.1.2.2 Regionprops function

Regionprops is a function found in MATLAB and used to measure segments. It approximates the lengths and widths by fitting an ellipse of best fit. The major and minor axes of the fitted ellipse are then measured, giving the length and width respectively. While it is used [114], it is not ideal. Problems arise when the segment is not suitably shaped to fit a close (or tight) ellipse (figure 5.4).

5.1.2.3 Yan's width measuring algorithm

Yan's width measuring algorithm [115] makes measurements by segmenting the image to determine the region(s) to be measured, which are then skeletalised. Spurs are removed and the 'centerline' (length line) divided into smaller regions to which second order polynomials are fit. The placement of the widths is then determined by finding the normal of the curve and deducing the first order polynomial equation.

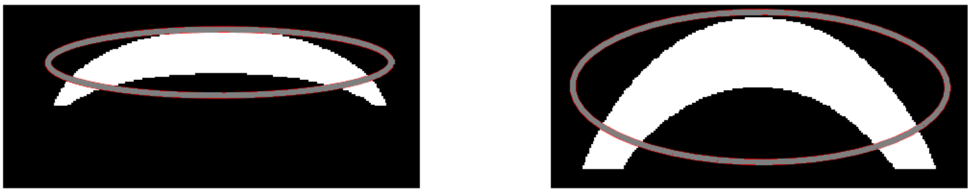


Figure 5.4: Placement and sizing of ellipse used by regionprops function to determine region length and width (imposed in grey).

5.1.3 Aims and Objectives

The aims and objectives of this chapter are:

- Develop an algorithm to measure the length and width of a segmented *S. pombe* cell that is more consistently accurate than existing methods.

5.2 Methodology

5.2.1 Previously described datasets and algorithms

This chapter uses the fisYeast datasets described in chapter 2.1. The Outline algorithm that was developed in chapter 4 is also used.

5.2.2 The CellGeometrics algorithm

The CellGeometrics algorithm is designed to measure the area, length and average width (mean of the central and quartile divisions of the proposed segmentation's length) in pixels. Figure 5.5 depicts the steps taken by the algorithm to achieve this.

CellGeometrics (figure 5.5) has the same input requirements as the OutlineVer3.2 algorithm: the input image, the image magnification, and the

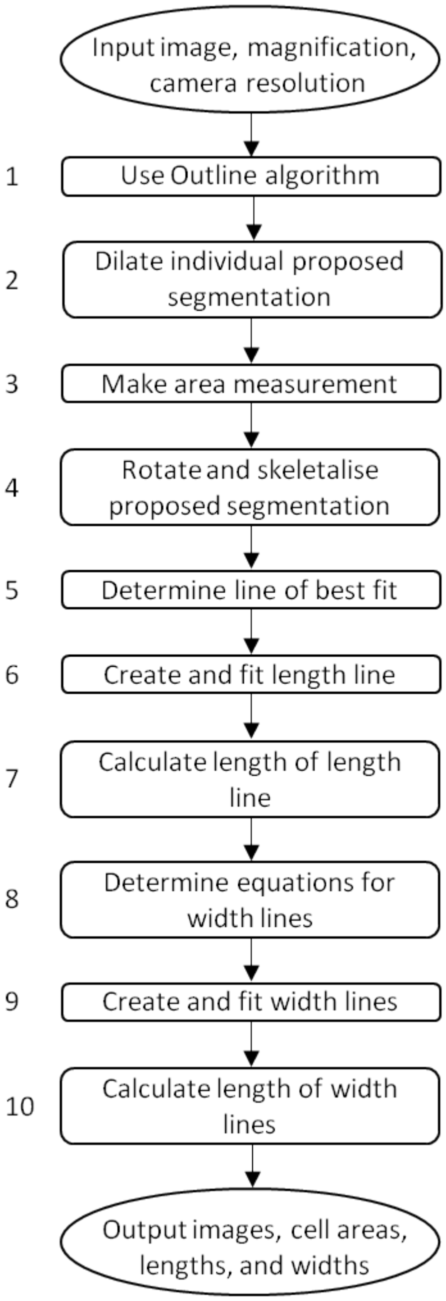

Input image, magnification, camera resolution
1
Use Outline algorithm
2
Dilate individual proposed segmentation
3
Make area measurement
4
Rotate and skeletalise proposed segmentation
5
Determine line of best fit
6
Create and fit length line
7
Calculate length of length line
8
Determine equations for width lines
9
Create and fit width lines
10
Calculate length of width lines
Output images, cell areas, lengths, and widths

Figure 5.5: Schematic of CellGeometrics.

resolution of the camera. These inputs are then run through OutlineVer3.2 (from chapter 4.2.2.3) with the final output image and the image matrix from step 5 of figure 4.6. The final output image is saved while each potential segmentation found in the image matrix is taken in turn, dilated (by a disk structural element with a radius of 15 (for the fisYeast dataset), and analysed geometric measurements.

The redilated potential segmentation is rotated so that the length is approximately aligned horizontally based on the angle of orientation given by the reigonprops function. The medial axis is then identified and any spurs removed/reduced (by 30 pixels) using the 'skel' and 'spur' functions from bwmorph.

Once the proposed segmentation has been transformed into a pixel thick line, a line of best fit is required. This is fitted using the polyfit function as both a first order and a second order polynomial. If the x^2 term is between 0.0005 and -0.0005, the proposed segmentation is treated as a straight cell and uses the first order polynomial equation. Otherwise, the cell is deemed to be bent and the second order polynomial equation is used.

The line of best fit is then plotted and cropped to the size of the proposed segmentation, making use of the polyval function. The ends of the line are determined and the length of the line or arc is calculated (using equation 5.1 if it is straight or equation 5.2 (where $f(x)$ is the equation of the second order polynomial) if it is not).

$$l = \sqrt{(y - y_1)^2 + (x - x_1)^2} \tag{5.1}$$

$$l = \int_{x}^{x_1} \sqrt{1 + \left(\frac{dy}{dx} f(x)\right)^2} dx \tag{5.2}$$

To determine the width of the proposed segmentation, the coordinates of the centre and quartiles of the length line are found. They are inserted into equation 5.3 (where $f(x)$ is the equation of the length line) to determine the gradient of the length line at that point. The gradient of the normal and the straight line constant is then calculated for each location (equations 5.4 and 5.5 respectively).

$$m = \frac{dy}{dx} f(x) \tag{5.3}$$

$$n = \frac{1}{m} \tag{5.4}$$

$$c = y - (m \cdot x) \tag{5.5}$$

After the components required to construct the straight line equations have been determined, the lines can be plotted and fitted to the region. The ends of these fitted lines are then used in equation 5.1 to calculate their length, giving the width of the proposed segmentation for three equally spaced points along the cell. The segmentation's outline, along with the length line and width lines are also saved as an output to allow the end user to assess their suitability.

5.2.3 Assessing the CellGeometrics algorithm

The performance of both the Outline algorithm and the novel measuring method are assessed. The results of the novel measuring method are also compared to the results of established measuring methods.

5.2.3.1 Performance of the Outline algorithm

The segmentations proposed by the Outline algorithm were assessed via the method described in chapter 2.5.1. Any cells that were not in focus enough to be considered for manual segmentation were not counted as potential cells for segmentation as any analysis of them would be inaccurate regardless of the method. If the Outline algorithm segments these cells it could skew analysis results, reducing the usefulness of the segmentation algorithm for the end-user.

5.2.3.2 Performance of the CellGeometrics algorithm

The images of the proposed segmentation with the length and width lines imposed on were visually assessed to determine several things. Firstly, if the length line is one of best fit, including if the cell should be considered as straight or bent. Secondly, if the position and angle of the width lines appears to be acceptable.

5.2.3.3 Comparison with other measuring techniques

The length and central width measurements of the proposed segmentations were compared to manually made measurements (averages from triplicate straight line measurements) as well as the measurements taken using the regionprops function on both the original proposed segmentation and the rotated segmentation (from step 4 of figure 5.5). The lengths were also compared to the pixel count of the medial axis provided by the skeletalisation of the proposed segmentation (from step 4 of figure 5.5). The medial axis was also found for the unrotated proposed segmentation for comparison (from step 2 of figure 5.5).

5.3 Results and Discussion

5.3.1 Development of the CellGeometrics algorithm

As stated in chapter 5.1, accurate measurements for the areas, lengths, and widths of proposed segmentations are important for biological studies. Determining the area of a proposed segmentation is simple and relies purely on the segmentation being accurate. Counting the number of pixels included in the proposed segmentation can be done using the 'area' command of the regionprops function or by asking for the number of pixels with a non-zero value if the segmentation is isolated. Finding the length or width of a cell is more difficult.

5.3.1.1 Length measurement method

To get an accurate length measurement of a segment, the measurement must follow any potential curvature through the middle of the segment, along the longest axis. The easiest way to identify the desired route is by finding the medial axis. This can be done using inbuilt MATLAB function bwmorph (using 'skel' with infinite iterations).

Rotating the segments to align horizontally is important for fitting a line or arc to the pixel coordinates for the medial axis (with the cell end spurs removed/reduced). *S. pombe* cells are typically considered straight, but many have a slight bend to them, especially some mutant strains. This means that first or second order polynomials are suitable; however, if the cell is orientated awkwardly, the line of best fit cannot be plotted well from the coordinates by a first or second order polynomial. Aligning the cell to the x axis ensures the most appropriate equation can be determined.

Having an equation for the length line means that it can be used to calculate accurate lengths with equations 5.1 and 5.2. To decide if a segmentation should be treated as though the cell were straight or bent is often an arbitrary decision. With CellGeometrics, the exact curvature required to define the difference can be changed by the end-user if they desire, yet it is consistent over any segment, image or dataset provided.

5.3.1.2 Width measurement method

One of the benefits of having an equation-defined length line is that width measurements can be made perpendicular to the length, regardless of the location along the medial axis. Being able to define the equation for width lines means that the actual width can be calculated and there is no uncertainty as to if the line measured is truly the width of the cell. Taking width measurements in this manner also means that multiple measurements can be made and their location within the cell can be set depending on the analysis required for the particular study.

5.3.1.3 Comparison to the Yan's width measuring algorithm

Yan's width measuring algorithm [115] uses similar techniques to CellGeometrics, but has three differences.

First, the removal of spurs is done differently by determining the longest continuous line and discarding any pixels that are not included in this route. This works well for pipes as they are reasonably symmetrical (reducing the likelihood of spurs) and extremely long. Problems would arise on cell segments as the ends of cells form split ends (figure 5.3) and the longest line would not maintain the appropriate trend.

Secondly, CellGeometrics does not divide the length line into smaller sections to fit a curve. This is necessary in Yan's algorithm as the segment is large and has multiple regions with different profiles; however, the cell segments do not have this problem as they do not have a point of inflection.

The third difference is that a quadratic equation is fitted for all length lines in Yan's algorithm [115] while CellGeometric uses a straight line equation if the x^2 term is between 0.0005 and -0.0005. The reason a straight line equation is used when a segment is classified as straight is because the line equation reflects the classification.

5.3.2 Assessing the CellGeometrics algorithm

Two things need to be considered to determine if the CellGeometrics algorithm is an improved method for measuring cell lengths and width: if the route used for the measurements are suitable, and if the measurements are closer to the manually collected measurements. After the success of the Outline algorithm is assessed, the measurement lines are checked and the numerical results compared.

5.3.2.1 Performance of the Outline algorithm

The fisYeast dataset was captured by an individual unrelated to this thesis. This means that the performance of the Outline algorithm is of interest as the imaging conditions match those of the 100x-oil-Z dataset, apart from the type of microscopy, but were not the images used in the development of the algorithm.

Of the 315 cells, there were: 101 true positive segmentations, 3 false positive segmentation, and 214 false negatives. This data produces an $F_{0.5}$ score of 0.6908. An $F_{0.5}$ score of 0.6908 is less than the 0.8529 (when J = 0.90) or 0.9291 (when J = 0.75) produced with the 100x-oil-Z dataset (table 4.11); however, the PombeX algorithm produces an $F_{0.5}$ score of 0.6782 on its own dataset (chapter 4.3.4.1). This means that the Outline algorithm performs comparably, on a truly random dataset, to the currently used available algorithm on their own dataset.

Several of the cells are not completely in focus (figures 5.6a-d) which can also affect the success of the segmentation algorithm. All of these cells can be manually segmented as a human can infer where the cell wall is, but computers cannot. The way figure 5.6c is focussed means that it resembles the phase-contrast images of the 100x-oil-Z dataset more closely than figure 5.6d does. Figure 5.6e shows three cells that have been successfully segmented, while figure 5.6f shows none. This suggests that the Outline algorithm is more successful on phase-contrast images then on transmitted light images.

5.3.2.2 Performance of the CellGeometrics algorithm

Visual assessment of the segmentation images overlaid with the length and width lines shows that of 104 segments, only four show poor length lines (figure 5.7). The furthest right image of figure 5.7 is the only poorly fitted straight line. Even the poorly fitted curved lines are not obscenely bad. None of the width lines appear incorrect.

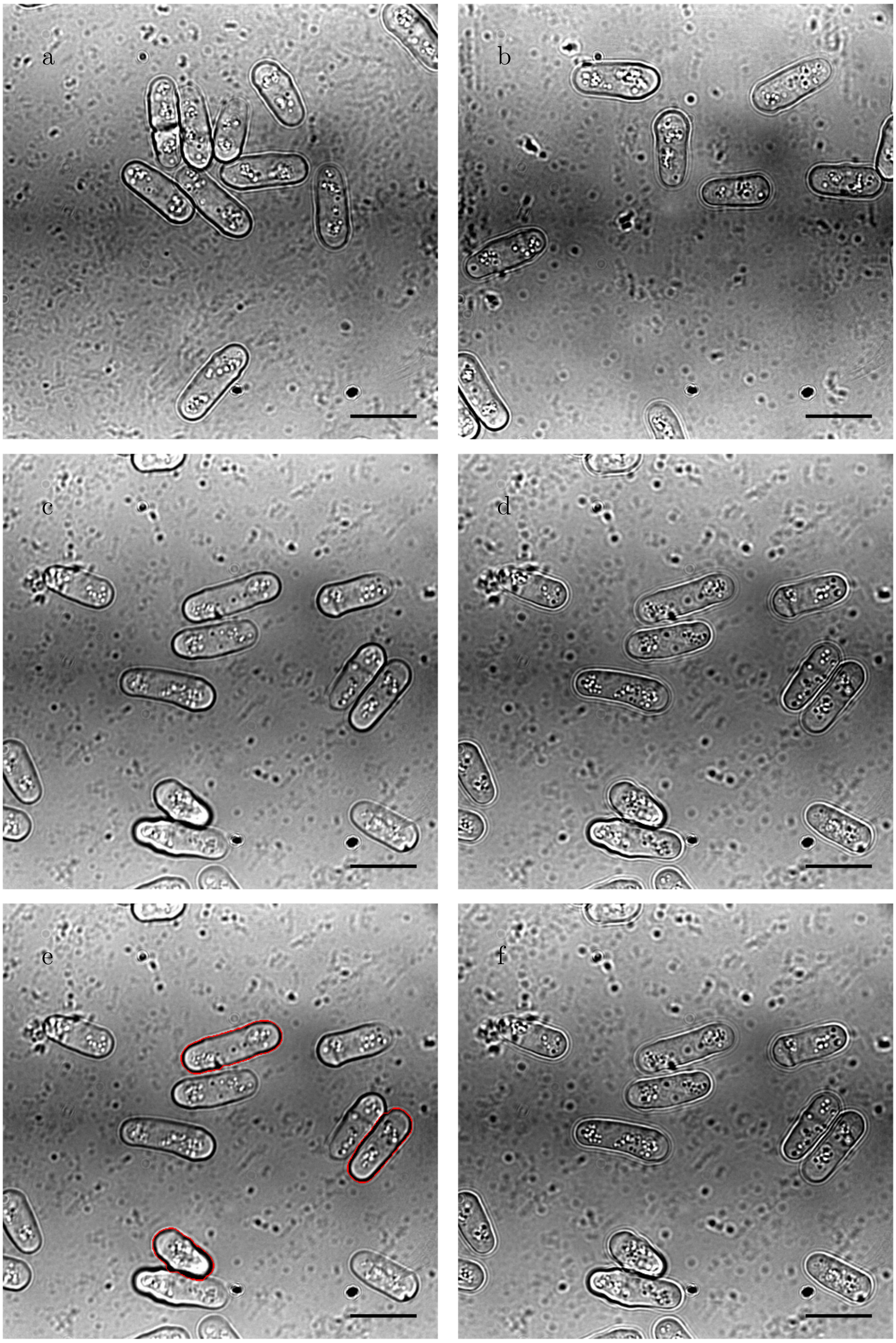


Figure 5.6: (a)-(d) images of the fisYeast dataset showing the focussing of the frames. (e) and (f) the output image of the Outline segmentation algorithm on images (c) and (d) respectively. Scale bars represent 10 µm.

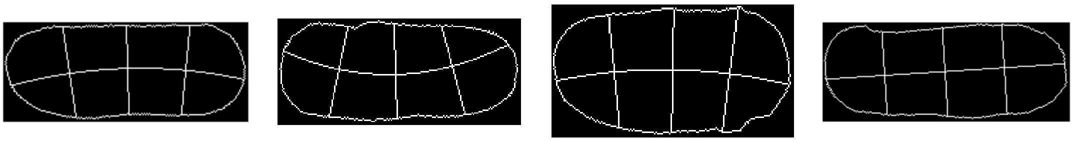


Figure 5.7: Poor length lines from CellGeometrics algorithm output. Images constructed of Outline algorithm segmentation, and the length line and width lines from CellGeometrics.

5.3.2.3 Comparison with other measuring techniques

When comparing the measurements of the CellGeometrics algorithm, measurements were taken from the proposed segmentations from the Outline algorithm. This ensures that the measurements are comparable as they all rely on the same input data instead of risking bias due to the performance of the Outline algorithm. Table 5.1 shows the mean length and width measurements produced by each measurement method.

Measurement method	**mean length**	**mean width**
Medial axis (unrotated)	12.83	n/a
Medial axis (rotated)	17.63	n/a
Regionprops (unrotated)	13.09	5.21
Regionprops (rotated)	13.09	5.21
CellGeometrics	12.27	4.55
Manual	12.32	4.75

Table 5.1: Mean length and width measurements (in µm, rounded to 2dp) for each measurement method.

Counting the number of pixels in the skeletalised segment (medial axis in table 5.1) is the least consistent, or useful, method. It is unuseful as it cannot provide a width measurement. It is inconsistent as the average length changes dramatically depending on the orientation of the segment. The average length of the rotated segments is over 5 µm larger than the average manual length measurement while for the unrotated segments it is comparable with the manual measurement (table 5.1). These differences occur due to the presence of spurs, which form during the skeletalisation process due to a lack of symmetry. Skeletalisation of regions causes problems as it rarely produces the desired output (also seen in [113]), therefore, counting the number of pixels in the medial axis is often flawed.

Manual measurements were made by selecting two points along the segmentation outline and the distance between the two points is calculated. This is the method used by many manual measurements programs, and similar to how the regionprops function measures the axes of the fitted ellipses. The fitting of the ellipse is where problems arise with this method. While the ellipses are fitted almost identically, the measurements taken for the length or width sometimes varies a minuscule amount if they are rotated or not (0.01 µm after converting from pixels and rounding, data not shown). These differences are negligible, not affecting the average for the dataset (table 5.1). In comparison to the manual measurements made, regionprops over estimates both measurements (table 5.1).

The measurement made by CellGeometrics are the closest to the manual measurements (table 5.1). The average width was 0.20 µm out while the average length was only 0.05 µm out. The reason why the width seems more disproportionate is likely due to human bias. Humans struggle to accurately identify the centre of a segment and determine the line that is perpendicular to the length. This leads to skewed width lines for measurements, which explains why CellGeometrics gave seven widths which were at least 1 µm (approximately 15 pixels) smaller than the manual measurements and none that were larger (data not shown).

As the manual measurements were all taken as straight lines, the length measurements were always going to vary slightly, especially when 50 of the 104 cells are classified as curved. There were only four lengths that differ by over 1 µm; however, two were larger in the manual measurement data compared to the CellGeometrics data, and two were smaller.

5.3.2.4 Comparison with manual analysis

In the previous subsections, only the cells segmented by the Outline algorithm were considered. This could be introducing a bias into the resulting data if the segmentation algorithm fails to identify a class of valid cells consistently (for example abnormally short or wide cells).

For [116], the researcher who captured the images also measured the areas and lengths of 16 cells from the dataset using MetaMorph [50]. The average length was calculated as 12.26 µm and the average apparent width (the average area divided by the average length) was 4.03 µm [unpublished data]. These measurements are lower than the manual measurements taken of all segments of the segmentation algorithm (12.32 µm average length and 4.75 µm average measured width; table 5.1); however, the measurements calculated by the CellGeometrics algorithm fall within both manual methods (12.27 µm average length and 4.55 µm average measured width; table 5.1).

The average length data does not suggest any bias in the Outline algorithm's segmentation for this dataset. The large range of width measurements is interesting. It shows the importance of generating an effective method for determining cell widths as apparent widths are not ideal for non-rectangular cells.

5.4 Conclusions

The aims and objectives of this chapter were:

- Develop an algorithm to measure the length and width of a segmented *S. pombe* cell that is more consistently accurate than existing methods.

CellGeometrics is an algorithm that can measure the lengths and widths of *S. pombe* cells. The lines used to measure the lengths appear to be suitable over

95% of the time while the widths are 100% visually accurate. The 'inappropriate' length lines still produce length measurements within 1 μm of the manual measurements, which means that it is not a critical failure for the algorithm. The numerical data show that the CellGeometrics algorithm produces measurements that are closer to the manual measurements than the other measuring techniques. Together with the fact that it is easy to modify the algorithm to take geometric measurements from other segmentation sources; this means that CellGeometrics successfully fulfils the aims and objectives of this chapter.

CHAPTER 6

Discussion and Conclusion

6.1 Ground Truth Datasets

The PombeX dataset [73] is easy to download, but is not easy to use. While there are 10 transmitted light microscopy images of *S. pombe* cells, there are no details about imaging conditions including about cell growth and mounting methods. There are also only nine ground truth segmentations which are named differently with a list assigning each ground truth image to the correct microscopy image. Upon checking that the images correspond correctly, it is clear that one of the ground truth images is duplicated and there are only eight images that have both a microscopy image and ground truth segmentation image. The method used to determine/acquire the ground truths is also unknown.

In [62], the percentage of cells segmented is defined as 'the percentage of cell contours with less than 10% pixel mismatch compared to gold standard cell contours, relative to all cells'. This is the same as calculating the Jaccard index for each and classifying cells as true positives when $J \geq 0.90$, which is comparable to the level of accuracy found from the manual segmentation experiment in chapter

3 (table 3.1). This supports the quantification method used for analysing the quality of the Outline segmentation algorithm.

6.2 Ground Truth Segmentation Algorithms

The four novel datasets with ground truth segments from chapter 3 are derived manually using the GTGen algorithm. This algorithm, along with its supporting algorithms, can be used by anyone to generate form-fitting ground truth segmentations for any image and compare corresponding proposed segmentations to assess segmentation quality.

Datasets are not limited to just microscopy images or those related to biological studies. Any type of image that may be segmented will benefit from ground truth segments as it allows for consistent measurements of segmentation quality for comparing segmentation algorithms. This is well established by the number of existing datasets available with ground truth segmentations [71, 72, 73] and programs allowing for 'easy' ground truth generation [74, 75, 76]; however, when it comes to closely following the contours of objects, these programs, and some datasets, are lacking.

The quantification method used in GTTest may not be the best suited for every type of segmentation problem. If a different method is desired, the algorithm can be modified. The code to reconstruct the ground truth segments will always be suitable, but the method used to identify corresponding segments in proposed segmentations may be inappropriate. This could be the case if a different quantification method is used or if the segmentation algorithm does not present segments which need to be dilated by a disk with a radius of 15 pixels.

6.3 Outline Segmentation Algorithm

Fully automated segmentation algorithms for fluorescent microscopy images are abundant [58, 117, 118, 119]. While this is extremely helpful for biological scientists, there is a wealth of data being lost as brightfield/phase-contrast microscopy images are ignored. Gaining this additional data could lead to unexpected deductions being made and help further research.

6.3.1 Segmentation limitations

There are many algorithms that try to exploit features in microscopy images to segment any type of cell; however, there are not enough unique features that are consistently present to design a singular, perfect segmentation algorithm. This means that some functions have to be used which might not be suitable for all cell types. The Outline algorithm (developed in chapter 4), is specifically designed for *S. pombe* and is limited by the use of two things: area-based filters, and erosions/dilations by disk-shaped structural elements. Even though the minimum and maximum areas for potential segmentations can be altered to suit the cell size, the use of erosions and dilations hinder the possible phenotypes that can be segmented by the algorithm. Any round, elliptical, or oblong shaped cells can be segmented providing they do not have delicate off-shoots such as flagella or hyphae. Organisms with irregular cell shapes will lose definition during the dilations and erosion.

Microscopy datasets show some variability in focus between images. This can have a major impact on exploitable features such as the level of contrast or perceived thickness of the cell wall (for *S. pombe*). To ensure that image acquisition bias did not influence the successfulness of the Outline algorithm, the fisYeast dataset was acquired for a study by a different researcher (chapter 2.4) [116] than the other datasets. The segmentation success rate is lower for the fisYeast dataset then anticipated due to the 100x magnification, immersion oil lens used; but, the type of microscopy changes from phase-contrast to transmitted light, which reduces the amount of data extractable during the steps before the morphological filters are used. Over 97% of the proposed segmentations made are accepted as true positives, which is important as minimising the number of false positive segmentations is vital in ensuring end-users benefit from using the algorithm.

6.3.2 Extracting segmentations for further analysis

One of the most important parts of image analysis is in the analysis of segmentations. This means that the proposed segmentations need to be available in some form to act as input data. Extracting the proposed segmentations from the output images of the PombeX program [62] is challenging due to the inclusion of black as one of the seven outlining colour. This means that over 14% of proposed segmentations cannot be used by other algorithms. Outline superimposes the proposed segmentations onto the contrast-enhanced input image in red. This means that the segmentation mask can be extricated by subtracting either the green or blue layers of the RGB output image from the red layer. The dilation of the outline of the proposed segmentation before it is superimposed means that the mask can be filled; however,as some of the segmentations touch or overlap, it is better if the pre-final dilation potential segmentations are exported and then dilated in turn. This is only suitable for some analysis methods and may cause more complications if the analysis algorithm is not amendable.

6.4 CellGeometrics Algorithm

Accurate geometric measurements are as important as accurate segmentations for researchers. The CellGeometrics algorithm was designed for measuring the lengths and widths of *S. pombe* cells, but can be used on a wider range of organisms/segments.

6.4.1 Limitations on lengths and widths algorithm

Quadratic or straight length lines can be measured using CellGeometrics. Cubic curves are not fitted, meaning that any segment with an 'S' shape is wrongly represented. Yan's width measuring algorithm [115] treats all length lines as curves, but breaks the medial axis into smaller segments to be either a first or second order polynomial. There are two problems with this method: where the division occurs along the length line affects the best-fit equation; and the use of the quadratic equation, even if the x^2 term almost 0, seems inappropriate. CellGeometrics determines if a cell is curved or if the x^2 constant is negligible and then uses the best fit equation for the type of cell given. It seems inappropriate to label a cell as straight but then treat it as though it was not to take measurements. Segments that produce short medial axes, such as circles, do not have enough data points to extrapolate a meaningful line of best fit. This is also true for third, or higher, polynomials of best fit, or segments with highly irregular shapes.

6.5 Impact of this research for bioscientists

The aim of the work carried out in this thesis was to improve/produce tools to assist bioscientists in their research. Two key considerations throughout the development of all the algorithms presented in this thesis were that the end products had to have high specificity and be user-friendly. Too many programs exist which do not meet these criteria, and many bioscientists do not have the knowledge to modify algorithms. This led to researchers not using the algorithms designed to help them.

In this thesis, an algorithm to segment phase contrast microscopy images of *S. pombe* cells and an algorithm to measure the length and width of provided segmentations are introduced. While the number of cells segmented is lower than ideal, the amount of time it takes for a researcher to capture more images is significantly less than the time it would take to manually measure the length and width of a smaller dataset. There are also programs that automate the capture of cells across a slide. This can further reduce the amount of time required of the researcher. The automated measurement algorithm is also more consistent in taking measurements in the same way, improving the reliability of measurements - including between different researchers and even across different laboratories/research groups.

The algorithm to generate ground truth segmentations provides a method for bioscientists to generate datasets with ground truths. These can then be used by people with scripting skills, but no specific biological knowledge, to develop algorithms that can then be used to assist bioscientists. Hopefully, this will result in more user-friendly algorithms and encourage automated analysis.

6.6 Further Work

The precision of the proposed segmentations generated by the GTGen algorithm relies heavily on human perception. Currently, the scale of the image used for manual segmentation is limited by display size. This means that large images are reduced in size to fit the screen, reducing the level of detail visible. Including the ability to zoom, and maintain accurate positioning/size of chosen knots, would improve the precision of ground truth segmentations.

Fluorescence microscopy images or time-lapse series could be incorporated with the Outline algorithm in an attempt to increase the number of proposed segmentations. Using accompanying microscopy images to allow for multiple pathways, or over-riding steps of the algorithm, during the segmentation process is not unheard of [55, 62]. The Outline segmentation algorithm could be improved by utilising additional inputs.

Only *S. pombe* cell segments were used in the development of CellGeometrics. This means that adapting it for segments that require third order polynomials or higher was not needed; however, some mutants or other organisms/segments would benefit from fitting cubic curves. More importantly, a more exacting method to determine the length line means less data points are lost during the removal of spurs. This leads to less extrapolations and assumptions.

6.7 Conclusions

The aims and objectives of this thesis were:

- Create a manual segmentation algorithm to generate ground truth segmentations.
- Define a method to use ground truth segmentations to quantify the success of segmentation algorithms.
- Develop an automated segmentation algorithm for phase contrast/transmitted light microscopy images of *S. pombe* cells.
- Determine a more accurate and precise method for measuring the length and width of segmented *S. pombe* cells.

In chapters 3, 4, and 5 we have achieved each of these goals. GTGen and GTVer allow for accurate, precise manual segmentation of any type of input images, while GTTest uses the Jaccard index along with a tolerance factor to identify correctly segmented segmentations before calculating an $F_{0.5}$ score. An $F_{0.5}$ score was chosen over an F_1 score as false positives are more of an issue than false negatives. The Outline segmentation algorithm outperforms the PombeX segmentation algorithm, and the CellGeometrics algorithm produces length and width measurements more consistent with manual measurements than other commonly used methods.

Bibliography

[1] C. R. Woese, O. Kandler, and M. L. Wheelis, “Towards a natural system of organisms: proposal for the domains Archaea, Bacteria, and Eucarya,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 87, pp. 4576–4579, Jun 1990.

[2] N. Iwabe, K. Kuma, M. Hasegawa, S. Osawa, and T. Miyata, “Evolutionary relationship of archaebacteria, eubacteria, and eukaryotes inferred from phylogenetic trees of duplicated genes,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 86, pp. 9355–9359, Dec 1989.

[3] T. Vellai and G. Vida, “The origin of eukaryotes: the difference between prokaryotic and eukaryotic cells,” *Proceedings of the Royal Society B: Biological Sciences*, vol. 266, pp. 1571–1577, Aug 1999.

[4] J. A. Leigh, S. Albers, H. Atomi, and T. Allers, “Model organisms for genetics in the domain Archaea: methanogens, halophiles, Thermococcales and Sulfolobales,” *FEMS Microbiology Reviews*, vol. 35, pp. 577–608, Jul 2011.

[5] T. Cavalier-Smith, “A revised six-kingdom system of life,” *Biological Reviews*, vol. 73, pp. 203–266, Aug 2007.

[6] G. Cooper, "The Eukaryotic Cell Cycle," in *The Cell: A Molecular Approach*, Sunderland (MA): Sinauer Associates, 2nd ed., 2000.

[7] A. B. Pardee, "A restriction point for control of normal animal cell proliferation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 71, pp. 1286–1290, Apr 1974.

[8] A. J. Waskiewicz and J. A. Cooper, "Mitogen and stress response pathways: Map kinase cascades and phosphatase regulation in mammals and yeast," *Current Opinion in Cell Biology*, vol. 7, pp. 798 – 805, Dec 1995.

[9] W. Zhang and H. T. Liu, "MAPK signal pathways in the regulation of cell proliferation in mammalian cells," *Cell Research*, vol. 12, p. 9, Mar 2002.

[10] X. Zhu, C. A. Rottkamp, H. Boux, A. Takeda, G. Perry, and M. A. Smith, "Activation of p38 kinase links Tau phosphorylation, oxidative stress, and cell cycle-related events in Alzheimer disease," *Journal of Neuropathology & Experimental Neurology*, vol. 59, pp. 880–888, Oct 2000.

[11] T. Yamada, H. Ishihara, A. Tamura, R. Takahashi, S. Yamaguchi, D. Takei, A. Tokita, C. Satake, F. Tashiro, H. Katagiri, H. Aburatani, J. Miyazaki, and Y. Oka, "WFS1-deficiency increases endoplasmic reticulum stress, impairs cell cycle progression and triggers the apoptotic pathway specifically in pancreatic β-cells," *Human Molecular Genetics*, vol. 15, pp. 1600–1609, May 2006.

[12] C. Callens, "A multiplex culture system for the long-term growth of fission yeast cells," *Yeast*, vol. 34, pp. 343–355, Aug 2017.

[13] G. Costello, L. Rodgers, and D. Beach, "Fission yeast enters the stationary phase G0 state from either mitotic G1 or G2," *Current Genetics*, vol. 11, pp. 119–125, Nov 1986.

[14] M. Werner-Washburne, E. Braun, G. C. Johnston, and R. A. Singer, "Stationary phase in the yeast *Saccharomyces cerevisiae*," *Microbiology and Molecular Biology Reviews*, vol. 57, pp. 383–401, Jun 1993.

[15] I. Watanabe and S. Okada, "Stationary phase of cultures mammalian cells (l5178y)," *The Journal of Cell Biology*, vol. 35, pp. 285–294, Nov 1967.

[16] J. Petersen and P. Russell, "Growth and the environment of *Schizosaccharomyces pombe*," in *Cold Spring Harbor Protocols*, vol. 2016, p. pdb.top079764, Mar 2016.

[17] T. Straube and C. Müller, "How to do a proper cell culture quick check," 2016. Available: https://www.leica-microsystems.com/science-lab/how-to-do-a-proper-cell-culture-quick-check/, Accessed: Sept 2017.

[18] A. Mazouzi, G. Velimezi, and J. I. Loizou, "DNA replication stress: causes, resolution and disease," *Experimental Cell Research*, vol. 329, pp. 85–93, Nov 2014.

[19] L. Wiesmüller, J. M. Ford, and R. H. Schiestl, "DNA damage, repair, and diseases," *Journal of Biomedicine and Biotechnology*, vol. 2, p. 45, Sept 2002.

[20] M. M. Ling and B. H. Robinson, "Approaches to DNA mutagenesis: an overview," *Analytical Biochemistry*, vol. 254, pp. 157–178, Dec 1997.

[21] M. Smith, "*In vitro* mutagenesis," *Annual Review of Genetics*, vol. 19, pp. 423–462, Dec 1985.

[22] P. Carter, "Site-directed mutagenesis.," *Biochemical Journal*, vol. 237, pp. 1–7, Jul 1986.

[23] Nobel Media, “The Nobel Prize in Chemistry 2008 - Press Release.” Available: https://www.nobelprize.org/prizes/chemistry/2008/press-release/, Accessed: Aug 2017.

[24] N. C. Shaner, G. H. Patterson, and M. W. Davidson, “Advances in fluorescent protein technology,” *Journal of Cell Science*, vol. 120, pp. 4247–4260, Jul 2007.

[25] Q. Ni, S. Mehta, and J. Zhang, “Live-cell imaging of cell signaling using genetically encoded fluorescent reporters,” *The FebS Journal*, vol. 285, pp. 203–219, Jan 2017.

[26] B. A. Pollok and R. Heim, “Using GFP in FRET-based applications,” *Trends in Cell Biology*, vol. 9, pp. 57 – 60, Feb 1999.

[27] C. S. Hoffman, V. Wood, and P. A. Fantes, “An ancient yeast for young geneticists: a primer on the *Schizosaccharomyces pombe* model system,” *Genetics*, vol. 201, pp. 403–423, Oct 2015.

[28] N. Maclean, “Electron microscopy of a fission yeast, *Schizosaccharomyces pombe*,” *Journal of Bacteriology*, vol. 88, pp. 1459–1466, Nov 1964.

[29] F. Chang and K. C. Huang, “How and why cells grow as rods,” *BMC Biology*, vol. 12, pp. 54–64, Aug 2014.

[30] V. Wood, R. Gwilliam, M.-A. RaJandream, M. Lyne, R. Lyne, A. Stewart, J. Sgouros, N. Peat, J. Hayles, S. Baker, D. Basham, S. Bowman, K. Brooks, D. Brown, S. Brown, T. Chillingworth, C. Churcher, M. Collins, R. Connor, A. Cronin, P. Davis, T. Feltwell, A. Fraser, S. Gentles, A. Goble, N. Hamlin, D. Harris, J. Hidalgo, G. Hodgson, S. Holroyd, T. Hornsby, S. Howarth, E. J. Huckle, S. Hunt, K. Jagels, K. James, L. Jones, M. Jones, S. Leather, S. McDonald, J. McLean, P. Mooney,

S. Moule, K. Mungall, L. Murphy, D. Niblett, C. Odell, K. Oliver, S. O'Neil, D. Pearson, M. A. Quail, E. Rabbinowitsch, K. Rutherford, S. Rutter, D. Saunders, K. Seeger, S. Sharp, J. Skelton, M. Simmonds, R. Squares, S. Squares, K. Stevens, K. Taylor, R. G. Taylor, A. Tivey, S. Walsh, T. Warren, S. Whitehead, J. Woodward, G. Volckaert, R. Aert, J. Robben, B. Grymonprez, I. Weltjens, E. Vanstreels, M. Rieger, M. Schäfer, S. Müller-Auer, C. Gabel, M. Fuchs, C. Fritzc, E. Holzer, D. Moestl, H. Hilbert, K. Borzym, I. Langer, A. Beck, H. Lehrach, R. Reinhardt, T. M. Pohl, P. Eger, W. Zimmermann, H. Wedler, R. Wambutt, B. Purnelle, A. Goffeau, E. Cadieu, S. Dréano, S. Gloux, V. Lelaure, S. Mottier, F. Galibert, S. J. Aves, Z. Xiang, C. Hunt, K. Moore, S. M. Hurst, M. Lucas, M. Rochet, C. Gaillardin, V. A. Tallada, A. Garzon, G. Thode, R. R. Daga, L. Cruzado, J. Jimenez, M. Sánchez, F. del Rey, J. Benito, A. Domínguez, J. L. Revuelta, S. Moreno, J. Armstrong, S. L. Forsburg, L. Cerrutti, T. Lowe, W. R. McCombie, I. Paulsen, J. Potashkin, G. V. Shpakovski, D. Ussery, B. G. Barrell, and P. Nurse, "The genome sequence of *Schizosaccharomyces pombe*," *Nature*, vol. 415, pp. 871–880, Feb 2002.

[31] V. Wood, M. A. Harris, M. D. McDowall, K. Rutherford, B. W. VAughan, D. M. Staines, M. Aslett, A. Lock, J. Bähler, P. J. Kersey, and S. G. Oliver, "PomBase: a comprehensive online resource for fission yeast," *Nucleic Acids Research*, vol. 40, pp. 695–699, Jan 2012.

[32] S. C. Li and P. M. Kane, "The yeast lysosome-like vacuole: endpoint and crossroads," Apr 2009.

[33] B. F. Johnson, B. Y. Yoo, and G. B. Calleja, "Cell division in yeasts: movement of organelles associated with cell plate growth of *Schizosaccharomyces pombe*," *Journal of Bacteriology*, vol. 115, pp. 358–366, Jul 1973.

[34] R. Imai, T. Nozaki, T. Tani, K. Kaizu, K. Hibino, S. Ide, S. Tamura, K. Takahashi, M. Shribak, and K. Maeshima, "Density imaging of heterochromatin in live cells using orientation-independent-DIC microscopy," Nov 2017.

[35] J. Selinummi, P. Ruusuvuori, I. Podolsky, A. Ozinsky, E. Gold, O. Yli-Harja, A. Aderem, and I. Shmulevich, "Bright field microscopy as an alternative to whole cell fluorescence in automated analysis of macrophage images," *PLOS ONE*, vol. 4, pp. 1–9, Oct 2009.

[36] A. Kruse, S. Fattah-Hosseini, S. Saha, R. Johnson, E. Warwick, K. Sturgeon, L. Mueller, M. J. MacCoss, R. G. Shatters, and M. Cilia Heck, "Combining 'omics and microscopy to visualize interactions between the Asian citrus psyllid vector and the Huanglongbing pathogen *Candidatus* Liberibacter asiaticus in the insect gut," *PLOS ONE*, vol. 12, Jun 2017.

[37] K. Cieśliński and J. Ries, "The yeast kinetochore — structural insights from optical microscopy," *Current Opinion in Chemical Biology*, vol. 20, pp. 1 – 8, Jun 2014.

[38] M. Leboffe and B. Pierce, "Microscopy," in *A Photographic Atlas for the Microbiology Laboratory* (D. Ferguson, ed.), pp. 31–36, Englewood CO: Morton Publishing, 4th ed., 2011.

[39] M. Wilson, "Immersion Objectives: Using Oil, Glycerol, or Water to Overcome some of the Limits of Resolution," 2017. Available: https://www.leica-microsystems.com/science-lab/immersion-objectives-using-oil-glycerol-or-water-to-overcome-some-of-the-limits-of-resolution/, Accessed: Apr 2018.

[40] J. R. Benford, “Numerical Aperture,” in *The Theory of the Microscope* (J. R. Benford, ed.), pp. 5–6, Rochester, NY: Bausch & Lomb Incorporated, 1st ed., 1960.

[41] G. Choi, H. Um, M. Kim, Y. Kim, H. Kang, B. Chung, and Y. Kim, “Isolation and characterization of ethanol-producing *Schizosaccharomyces pombe* CHFY0201,” *Journal of Microbioloy and Biotechnology*, vol. 20, pp. 828–834, Apr 2010.

[42] K. Baker, S. Kirkham, L. Halova, J. Atkin, M. Franz-Wachtel, D. Cobley, K. Krug, B. Maček, D. P. Mulvihill, and J. Petersen, “TOR complex 2 localises to the cytokinetic actomyosin ring and controls the fidelity of cytokinesis,” *Journal of Cell Science*, vol. 129, pp. 2613–2624, Jul 2016.

[43] F. K. Khasanov, A. F. Salakhova, O. S. Khasanova, A. L. Grishchuk, O. V. Chepurnaja, V. G. Korolev, J. Kohli, and V. I. Bashkirov, “Genetic analysis reveals different roles of *Schizosaccharomyces pombe* sfr1/dds20 in meiotic and mitotic DNA recombination and repair,” *Current Genetics*, vol. 54, p. 197, Sept 2008.

[44] Nobel Media, “The Nobel Prize in Physics 1965 - Summary.” Available: https://www.nobelprize.org/prizes/physics/1953/summary, Accessed: May 2019.

[45] F. C. Grigg, “Colour-contrast phase microscopy,” *Nature*, vol. 165, p. 368, Mar 1950.

[46] B. Chazotte, “Mounting live cells onto microscope slides,” in *Cold Spring Harbor Protocols*, vol. 2011, p. pdb.prot5554, 2011.

[47] R&D Systems, “Immunocytochemistry (ICC) Protocols,” 2017. Available: https://www.rndsystems.com/protocol-types/immunocytochemistry-icc, Accessed: Nov 2017.

[48] D. P. Mulvihill, “Live Cell Imaging in Fission Yeast,” in *Fission Yeast: A Laboratory Manual* (I. Hagan, A. Carr, A. Grallert, and P. Nurse, eds.), pp. 220–222, Cold Spring Harbor, NY: CSH Press, 1st ed., 2016.

[49] J. M. Perkel, “How to choose a microscopy camera,” 2015. Available: https://www.biocompare.com/Editorial-Articles/172456-How-to-Choose-a-Microscopy-Camera/, Accessed: Nov 2017.

[50] Molecular Devices, “MetaMorph microscopy automation and image analysis software.” Available: https://www.moleculardevices.com/products/cellular-imaging-systems/acquisition-and-analysis-software/metamorph-microscopy#gref, Accessed: Feb 2016.

[51] Zeiss, “Zeiss Zen - digital imaging for light microscopy.” Available: https://www.zeiss.com/microscopy/int/products/microscope-software/zen.html, Accessed: Apr 2018.

[52] Spot Imaging, “Spot basic image capture software.” Available: http://www.spotimaging.com/software/spot-basic/, Accessed: Apr 2018.

[53] J. F. Dorn, G. Danuser, and G. Yang, “Computational processing and analysis of dynamic fluorescence image data,” in *Fluorescent proteins*, vol. 85 of *Methods in Cell Biology*, pp. 497–538, Academic Press, 2008.

[54] J. Schindelin, C. T. Rueden, M. C. Hiner, and K. W. Eliceiri, “The ImageJ ecosystem: an open platform for biomedical image analysis,” *Molecular Reproduction and Development*, vol. 82, pp. 518–529, Jul 2015.

[55] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat, P. Golland, and D. M. Sabatini, “CellProfiler: image analysis software

for identifying and quantifying cell phenotypes," *Genome Biology*, vol. 7, p. 100, Oct 2006.

[56] C. Mayer, S. Dimopoulos, F. Rudolf, and J. Stelling, "Using CellX to quantify intracellular events," *Current Protocols in Molecular Biology*, vol. 101, pp. 1–20, Jan 2013.

[57] S. Dimopoulos, C. E. Mayer, F. Rudolf, and J. Stelling, "Accurate cell segmentation in microscopy images using membrane patterns," *Bioinformatics*, vol. 30, no. 18, pp. 2644–2651, 2014.

[58] E. Hodneland, T. Kögel, D. M. Frei, H. Gerdes, and A. Lundervold, "CellSegm - a MATLAB toolbox for high-throughput 3D cell segmentation," *Source Code for Biology and Medicine*, vol. 8, p. 16, Aug 2013.

[59] F. D. Sigoillot, J. F. Huckins, F. Li, X. Zhou, S. T. C. Wong, and R. W. King, "A time-series method for automated measurement of changes in mitotic and interphase duration from time-lapse movies," *PLOS ONE*, vol. 6, pp. 1–14, Sept 2011.

[60] Y. Chang, H. Yokota, K. Abe, C. Tang, and M. Tasi, "Automated detection and tracking of cell clusters in time-lapse fluorescence microscopy images," *Journal of Medical and Biological Engineering*, vol. 37, pp. 18–25, Feb 2017.

[61] C. Versari, S. Stoma, K. Batmanov, A. Llamosi, F. Mroz, A. Kaczmarek, M. Deyell, C. Lhoussaine, P. Hersen, and G. Batt, "Long-term tracking of budding yeast cells in brightfield microscopy: CellStar and the Evaluation Platform," *Journal of the Royal Society Interface*, vol. 14, Feb 2017.

[62] J. Peng, "PombeX: robust cell segmentation for fission yeast transillumination images," *PLOS ONE*, vol. 8, p. e81434, Dec 2013.

[63] F. Mualla, S. Schöll, B. Sommerfeldt, A. Maier, and J. Hornegger, "Automatic cell detection in bright-field microscope images using SIFT, random forests, and hierarchical clustering," *IEEE Transactions on Medical Imaging*, vol. 32, pp. 2274–2286, Aug 2013.

[64] G. Sarker, S. Dhua, and M. Besra, "An optimal clustering for fuzzy categorization of cursive handwritten text with weight learning in textual attributes," in *IEEE 2nd International Conference on Recent Trends in Information Systems*, pp. 6–11, July 2015.

[65] A. Gacsádi and P. Szolgay, "Variational computing based segmentation methods for medical imaging by using CNN," in *12th International Workshop on Cellular Nanoscale Networks and their Applications*, Mar 2010.

[66] M. Strzelecki, J. Lee, and S. H. Jeong, "Analysis of biomedical textured images with application of synchronized oscillator-based CNN," in *12th International Workshop on Cellular Nanoscale Networks and their Applications*, Mar 2010.

[67] MathWorks, "Image processing techniques for image analysis." Available: https://uk.mathworks.com/discovery/image-analysis.html, Accessed: May 2017.

[68] MATLAB, *version 9.0 (R2016a).* Natick, Massachusetts, USA: The MathWorks Inc., 2016.

[69] H. Fleyeh, "Color detection and segmentation for road and traffic signs," in *IEEE Conference on Cybernetics and Intelligent Systems*, vol. 2, pp. 809–814, Dec 2004.

[70] S. Moreno, A. Klar, and P. Nurse, "Molecular genetic analysis of fission yeast *Schizosaccharomyces pombe*," in *Guide to Yeast Genetics and*

Molecular Biology, vol. 194 of *Methods in Enzymology*, pp. 795 – 823, Academic Press, Dec 1990.

[71] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *Proc. 8th Int’l Conf. Computer Vision*, vol. 2, pp. 416–423, July 2001.

[72] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman, “Ground truth dataset and baseline evaluations for intrinsic image algorithms,” in *IEEE 12th International Conference on Computer Vision*, pp. 2335–2342, Oct 2009.

[73] C. Zhang, “2D bright field Fission yeast cell images with ground truth annotations,” Mar. 2017. Available: https://doi.org/10.5281/zenodo.344880, Accessed: May 2017.

[74] D. Doermann, E. Zotkina, and H. Li, “GEDI–a groundtruthing environment for document images,” Jan 2010.

[75] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “LabelMe: a database and web-based tool for image annotation,” *International Journal of Computer Vision*, vol. 77, pp. 157–173, May 2008.

[76] P. A. Yushkevich, J. Piven, H. Cody Hazlett, R. Gimpel Smith, S. Ho, J. C. Gee, and G. Gerig, “User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability,” *Neuroimage*, vol. 31, pp. 1116–1128, Mar 2006.

[77] R. Delgado-Gonzalo, N. Chenouard, and M. Unser, “Spline-based deforming ellipsoids for interactive 3D bioimage segmentation,” *IEEE Transactions on Image Processing*, vol. 22, pp. 3926–3940, May 2013.

[78] A. A. Godil, R. D. Eastman, and T. H. Hong, "Ground truth systems for object recognition and tracking," *Journal of Research of NIST*, Mar 2013.

[79] A. Lehmussola, P. Ruusuvuori, J. Selinummi, H. Huttunen, and O. Yli-Harja, "Computational framework for simulating fluorescence microscope images with cell populations," *IEEE Transactions on Medical Imaging*, vol. 26, pp. 1010–1016, Jul 2007.

[80] A. Lehmussola, P. Ruusuvuori, J. Selinummi, T. Rajala, and O. Yli-Harja, "Synthetic images of high-throughput microscopy for validation of image analysis methods," *Proceedings of the IEEE*, vol. 96, pp. 1348–1360, Aug 2008.

[81] S. Rajaram, B. Pavie, N. E. F. Hac, S. J. Altschuler, and L. F. Wu, "SimuCell: a flexible framework for creating synthetic microscopy images," *Nature Methods*, vol. 9, p. 634, Jun 2012.

[82] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: an empirical evaluation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2294–2301, Jun 2009.

[83] K. McGuinness and N. E. O'Connor, "A comparative evaluation of interactive segmentation algorithms," *Pattern Recognition*, vol. 43, no. 2, pp. 434 – 444, 2010.

[84] D. Crevier, "Image segmentation algorithm development using ground truth image data sets," *Computer Vision and Image Understanding*, vol. 112, pp. 143 – 159, Nov 2008.

[85] T. Malisiewicz and A. Efros, "Improving spatial support for objects via multiple segmentations," *Proceedings of the British Machine Vision Conference*, Jan 2007.

[86] S. Takemoto, S. Yoshizawa, Y. Tsujimura, and H. Yokota, "A novel performance evaluation system for fluorescent cell image segmentation," in *First International Symposium on Computing and Networking*, pp. 294–299, Dec 2013.

[87] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 929–944, Jun 2007.

[88] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical Association*, vol. 66, pp. 846–850, Dec 1971.

[89] M. Polak, H. Zhang, and M. Pi, "An evaluation metric for image segmentation of multiple objects," *Image and Vision Computing*, vol. 27, pp. 1223 – 1227, Jul 2009.

[90] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," *Proceedings Eighth IEEE International Conference on Computer Vision*, vol. 2, Jul 2001.

[91] P. Ruusuvuori, T. Äijö, S. Chowdhury, C. Garmendia-Torres, J. Selinummi, M. Birbaumer, A. M. Dudley, L. Pelkmans, and O. Yli-Harja, "Evaluation of methods for detection of fluorescence labeled subcellular objects in microscope images," *BMC Bioinformatics*, vol. 11, p. 248, May 2010.

[92] D. Powers, "Evaluation : from precision , recall and F-measure to ROC, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, pp. 37–63, Dec 2011.

[93] Q. Huang and B. Dom, "Quantitative methods of evaluating image segmentation," in *Proceedings., International Conference on Image Processing*, vol. 3, pp. 53–56 vol.3, Oct 1995.

[94] J. O'Brien, S. Hoque, D. P. Mulvihill, and K. Sirlantzis, "Novel segmentation strategies for fully-automated analysis of yeast images," 2014.

[95] S. Boker, M. Neale, H. Maes, M. Wilde, M. Spiegel, T. Brick, J. Spies, R. Estabrook, S. Kenny, T. Bates, P. Mehta, and J. Fox, "OpenMx: an open source extended structural equation modeling framework," *Psychometrika*, Apr 2011.

[96] "Ground-truth data cannot do it alone," *Nature Methods*, vol. 8, p. 885, Nov 2011.

[97] S. Rajaram, B. Pavie, N. E. F. Hac, S. J. Altschuler, and L. F. Wu, "SimuCell: a flexible framework for creating synthetic microscopy images," *Nature Methods*, vol. 9, p. 634, Jun 2012.

[98] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, Jan 1979.

[99] G. Deng and L. W. Cahill, "An adaptive Gaussian filter for noise reduction and edge detection," in *IEEE Conference Record Nuclear Science Symposium and Medical Imaging Conference*, pp. 1615–1619 vol.3, Oct 1993.

[100] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679–698, Nov 1986.

[101] L. Yang, X. Wu, D. Zhao, H. Li, and J. Zhai, "An improved Prewitt algorithm for edge detection based on noised image," in *4th International Congress on Image and Signal Processing*, vol. 3, pp. 1197–1200, Oct 2011.

[102] W. Gao, X. Zhang, L. Yang, and H. Liu, "An improved Sobel edge detection," in *3rd International Conference on Computer Science and Information Technology*, vol. 5, pp. 67–71, Jul 2010.

[103] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Medical Image Computing and Computer-Assisted Intervention* (W. M. Wells, A. Colchester, and S. Delp, eds.), pp. 130–137, Jun 1998.

[104] T. Li, X. Zhang, and C. Li, "An improved adaptive image filter for edge and detail information preservation," in *International Conference on Systems and Informatics*, pp. 1870–1873, May 2012.

[105] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *International Journal of Computer Vision*, vol. 1, pp. 321–331, Jan 1988.

[106] C. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, pp. 359–369, Mar 1998.

[107] S. Beucher and C. Lantuéjoul, "Use of watersheds in contour detection," *In International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation*, vol. 132, Jan 1979.

[108] R. Adams and L. Bischof, "Seeded region growing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 641–647, Jun 1994.

[109] F. Meyer, "The watershed concept and its use in segmentation : a brief history," *Computer Vision and Pattern Recognition*, Feb 2012.

[110] J. Y. Peng, Y. J. Chen, M. D. Green, S. L. Forsburg, and C. N. Hsu, "Robust cell segmentation for *Schizosaccharomyces pombe* images with focus gradient," in *IEEE 10th International Symposium on Biomedical Imaging*, pp. 414–417, Apr 2013.

[111] M. Wu, R. F. Fraser, and C. W. Chen, "A novel algorithm for computer-assisted measurement of cervical length from transvaginal ultrasound images," *IEEE Transactions on Information Technology in Biomedicine*, vol. 8, pp. 333–342, Sept 2004.

[112] D. Wang, R. Lagerstrom, C. Sun, L. Bishof, P. Valotton, and M. Götte, "HCA-vision: Automated neurite outgrowth analysis," *Journal of Biomolecular Screening*, vol. 15, pp. 1165–1170, Oct 2010.

[113] M. Pool, J. Thiemann, A. Bar-Or, and A. E. Fournier, "NeuriteTracer: a novel ImageJ plugin for automated quantification of neurite outgrowth," *Journal of Neuroscience Methods*, vol. 168, pp. 134 – 139, Feb 2008.

[114] Y. J. Chen, M. D. Green, S. A. Sabatinos, S. L. Forsburg, C.-N. Hsu, and J. Y. Peng in *IEEE International Symposium on Bioelectronics and Bioinformatics*, Apr.

[115] C. Yan, D. Xu, Y. Liu, and F. Zhang, "Width estimation of glue line for engine oil sump based on planar vision," in *Chinese Control and Decision Conference*, pp. 2816–2821, May 2016.

[116] C. A. Johnson, H. R. Brooker, I. Gyamfi, J. O'Brien, B. Ashley, J. E. Brazier, A. Dean, J. Embling, E. Grimsey, A. C. Tomlinson, E. G. Wilson, M. A. Geeves, and D. P. Mulvihill, "Temperature sensitive point mutations in fission yeast tropomyosin have long range effects on the stability and function of the actin-tropomyosin copolymer," *Biochemical and Biophysical Research Communications*, Nov 2017.

[117] S. Blanchoud, D. Nicolas, B. Zoller, O. Tidin, and F. Naef, "CAST: an automated segmentation and tracking tool for the analysis of transcriptional kinetics from single-cell time-lapse recordings," *Methods*, vol. 85, pp. 3 – 11, Sept 2015.

[118] J. Kong, F. Wang, G. Teodoro, Y. Liang, Y. Zhu, C. Tucker-Burden, and D. J. Brat, "Automated cell segmentation with 3D fluorescence miscroscopy images," in *Proceedings / IEEE International Symposium on Biomedical Imaging: from nano to macro. IEEE International Symposium on Biomedical Imaging*, vol. 2015, pp. 1212–1215, Apr 2015.

[119] G. Herberich, T. Würflinger, A. Sechi, R. Windoffer, R. Leube, and T. Aach, "Fluorescence microscopic imaging and image analysis of the cytoskeleton," in *Conference Record of the Forty Fourth AsiloMar Conference on Signals, Systems and Computers*, pp. 1359–1363, Nov 2010.