Image Analysis of *Bacillus thuringiensis*

Gregory John Dickason BSc (Eng) (Cape Town)

Submitted to the University of Cape Town in fulfillment of the

requirements for the degree of Masters of Science in Engineering.

Bioprocess Engineering Research Group
Department of Chemical Engineering
University of Cape Town
Rondebosch, South Africa

April 1998

ACKNOWLEDGEMENTS

I would like to thank:

- *Dr. Caroline Henwood for her support, friendship and helpful criticism.*
- *My Supervisor, Assoc.Professor Susan Harrison, for her valuable support and advice.*
- *Dr. Mehdi Nemati for his approachability and help.*
- *Andrew Robinson for being (as always), the one to rely on.*
- *Sue Jobson for her organisation and cheerfulness.*
- *The "Bio's" group for friendship, fun and making varsity a fun place to be.*
- *The staff of the EMU unit of the University of Cape Town, particularly Dane. Gerneke, for loving what they do and inspiring others to do so too.*
- *Sashnee "Raja B" for help with the lab and for being such fun*
- *Naheed for starting me off in this exciting field*
- *Staff and postgrads of Chemical Engineering.*
- *AECI for their financial support for the project.*
- *Dr Andrew Bailey of AECI for his advice and help with Bt.*
- *My wife, Tracy, for her help, support and love through long nights and days.*

SYNOPSIS

This thesis concerns the development of a method to quantify the morphology of the bacterium *Bacillus thuringiensis,* and to automatically count the bacteria. The need to quantify the bacterial morphology arose out of the possibility of controlling a fermentation based on the morphology of the observed bacteria. Automatic counting of bacteria was considered necessary to reduce the inaccuracies that resulted in manual counts performed by different people.

Bacillus thuringiensis is a spore forming, gram-positive bacterium, which produces both intracellular spores and insecticidal protein crystals. The production of the insecticidal protein crystal makes *Bacillus thuringiensis* important as a producer of biological insecticides.

Automatic counting was developed in a Thoma counting chamber (Webber Scientific) at 200× magnification under dark field illumination. It was found that at this magnification the problem of out of focus cells was eliminated. The use of a thick coverslip, which reduces variability in slide preparation, was also possible at 200× magnification as the focal depth of the 20× objective lens was considerably larger than the 100× objective lens and thus the 20× objective lens could focus through the thick coverslip (20× objective lens with 10× magnification in eyepiece = 200× magnification). An automatic algorithm to acquire images was developed and 5 images per sample were acquired. Processing of the images involved automatically thresholding and then counting the number of bright objects in the image. Processing was thus rapid and the processing of the five images took no more than a few seconds.

Results showed that the correlation between the automatic and manual counts was good and that the use of a thick coverslip reduced variability in slide preparation. It was shown that the manual counting procedure, which necessarily used a thin coverslip at 1000× magnification, underestimated the volume of the Thoma counting chamber. This was a result of warping in the thin coverslip.

Morphological quantification and the identification of spores and protein crystals was performed on samples prepared by mounting them on water agar. This technique was developed to enable the user to ensure that all bacteria in an image are in focus and correctly orientated with respect to the objective lens of the microscope.

Thirty images per sample were automatically acquired for morphological measurements. The choice of this number of images was validated by a statistical analysis of up to 90 images. In this analysis it was shown that the standard deviation in the mean length reported decreased from 1.33 μm to 0.54 μm to 0.33 μm when 10, 30 and 90 images were analysed respectively. 30 images were thus shown to be a good compromise between accuracy in the measurements and the time taken to process the images.

Image storage was initially on hard drive, which necessitated the compression of the images. The use of the JPEG lossy compression method was shown to be adequate for the given system as the data extracted did not significantly differ when images stored in either JPEG or TIFF format were processed. Later images were stored as TIFF images as the acquiring of a CD-writer increased the available memory storage capabilities of the system. The storage of images in the TIFF format could have advantages in that uncompressed images would be available for additional processing techniques developed in the future.

Image analysis involved a sequence of steps, which were automated in a custom designed macro. Both gray scale and binary processing were performed. Templates were constructed and used to identify spores, cells and proteins and their relative positions in an image. The problem of overlapping cells was successfully overcome by applying the watershed separation technique to the inverted distance transform of the image.

The results showed that over 90% of cells and spores were found and processed correctly. Correct protein crystal identification was of the order of 80% as it relied on the correct identification of both cells and spores. Interesting trends in cell length were observed through shake flask and fermenter trials.

The quantification of cell shape was also investigated through the application of the Hough transform. It was found that this method could give better length and breadth measurements as it is based on approximating the bacterial shape as 2 circles joined by parallel lines.

A method to measure bacterial motility, both individually and in a whole image, was also attempted. The results indicate that individual bacterial motility appears fairly constant but that overall motility decreases with increased residence in the counting chamber. This is thought to be a result of bacterial attachment to the glass surfaces of the counting chamber.

In conclusion, a robust method to quantify the morphology of *Bacillus thuringiensis* has been developed. This can now be employed in studies to analyse morphological trends as a function of fermentation conditions. In addition, a method to automatically count bacteria in a counting chamber, using a thick coverslip, has been developed. This should both decrease the time taken to obtain a count and increase the reliability of any counts obtained. Future research into the development of methods to quantify the motility of the bacteria and also to directly quantify bacterial shape have also been discussed and initial work in these fields has been reported.

LIST OF TABLES

LIST OF FIGURES

GLOSSARY

4-connected Neighbours of a pixel are the adjacent pixels in the vertical and horizontal directions.

8-connected Neighbours of a pixel are all surrounding pixels (i.e. in the vertical, horizontal and diagonal directions)

Aerotaxis Movement toward or away from higher gas concentrations (e.g. around small bubbles)

Aspect Ratio The ratio of the dimensions of a pixel in the x and y directions.

Arithmetic coding A lossless coding scheme where an entire sequence of source symbols is assigned a single arithmetic code word.

Background The region in a binary image which is not of interest. (Often represented as black)

Binary Closing A binary dilation followed by a binary erosion.

Binary Dilation A binary operation where all background pixels connected to a foreground pixel are replaced by a foreground pixel (connection depending on definition as either 4- or 8- connected). This normally fills small holes on the boundary of an object.

Binary Erosion A binary operation where all foreground pixels connected to a background pixel are replaced by a background pixel (connection depending on definition as either 4- or 8- connected) This normally removes tendrils and small objects from the image.

Binary Fill A binary operation in which any background area which is completely surrounded by a foreground area is converted to a foreground area.

Binary Image Image with only two pixel values. 0, corresponding to black and 1, corresponding to white.

Binary Opening A binary erosion followed by a binary dilation. This can split finely joined objects.

Binary Outline A binary operation in which only the outermost pixels of a foreground object remain as foreground.

Bioprocess Any process that employs biological processes to perform chemical reactions.

Bioreactor The vessel in which biological reactions occur.

Chamfer 5-7-11 Distance approximation for digital images (Figure 3.11).

Chemotaxis Movement toward or away from a chemical.

Colony Forming Units (CFU's) The number of units, i.e. single cells or clumps of cells, which give rise to colonies in the dilution count technique (Barnett, 1992)

δ-endotoxin The protein crystal produced by *B. thuringiensis*.

DCT Discrete Cosine Transform, a method of transforming images from the spatial to the frequency domains.

Dilution plate count technique A technique to count cells by spreading a diluted sample across an agar plate and counting the number of colonies that form after incubation

Distance Transform	A transform that replaces a pixel's value with the value of the distance between the pixel and the nearest background pixel. The value is usually the Chamfer 5-7-11 distance.
Endospore	A differentiated cell formed within the cells of certain gram-positive bacteria. It is and extremely resistant to heat as well as to other harmful agents.
Endotoxin	A toxin not released from the cell; bound to the cell surface or intracellular.
Exotoxin	A toxin released extracellularly.
Exosporium	The region in the cell surrounding the spore.
Fermentation	Any process that produces a useful product by the mass culture of microorganisms
Fermenter	A vessel in which microorganisms can grow under sterile
Ferret Length	Maximum distance between parallel tangents to the object.
Flagella	A bacterial organ of motility.
Foreground	The region in a binary image which is of interest. (Often represented as white).
Frequency Domain	Rendition of an image into its component waves of varying amplitude and frequency (sum of all waves is the original image).
Geodesic Length	The length of an object corresponding to the shortest distance between two points on the objects perimeter.

Gram Stain	A differential staining procedure which divides bacteria into gram-positive and gram-negative groups based on their ability to retain crystal violet when decolorised with an organic solvent such as ethanol.
Gray Scale	Gray images, usually with 255 levels of gray value (8 bits per pixel), ranging from 0 (black) to 255 (white).
Homotopic	Two sets X and Y (or areas X and Y) are homotopic if X can be superimposed onto Y by means of continuous deformations. In this case a simply connected set will be transformed into a simply connected set, a set with one hole into a set with one hole, etc.
Hough transform	A transform whereby a probability matrix of the possible geometric features in an image is constructed and used to identify those features. The features are normally straight lines or circular arcs as the dimension of the probability matrix is dependant on the number of co-ordinates needed to uniquely mathematically decribe the object (line =2, circle =3)
Huffman Coding	A lossless variable length coding compression scheme.
Image Analysis	The analysis of scene data for the extraction of information about objects or regions of interest. This includes any image processing required.
Image Processing	The processing of images to highlight objects or regions of interest.
Innoculum	The microbial population introduced into a nutrient medium
Lepidoptera	An insect genus.
Lossless Compression	Compression of an image which is invertable. There is thus no potential loss of information.

Lossy Compression	Non-invertable compression of an image. Information may thus be lost in compression but compression ratios tend to be much higher than for lossless compression.
Mathematical Morphology	The examination of the geometrical structure of an image by matching it with small patterns at various locations in the image (Heijmans, 1995).
Mycelia	The term for a collection of hyphae
Pattern Matching	A procedure to characterise the the perimeter of an area as a sequence of lines and arcs. The procedure requires unbroken boundaries.
Phototaxis	Movement toward light.
Prolate Spheroid	A spheroid for which $a < c$ where $c \equiv \alpha(1+\varepsilon)$. Thus a prolate spheroid is "egg shaped"
Sobel Filter	A filter which replaces a pixel with the magnitude of the rate of change of gray level in the x and y directions as defined : $g(x,y) = \sqrt{G_x^2 + G_y^2}$. The Sobel filter highlights edges.
Spatial Domain	The aggregate of pixels which compose an image. Spatial domain methods are methods which act on these pixels.
Sporulation	The formation of spores
Threshold	The gray value that distinguishes between background and objects
Water Parting	Another term for a watershed

Watershed A region on a gray scale image or the distance transform of a binary image which corresponds to the crest lines between minima. It is used in the segmentation of images and also has applications in the separation of connected areas.

Watershed area The region surrounded by and defined by a watershed.

NOMENCLATURE

C	Circularity of an object (Perimeter^2/Area)
G_x and G_y	Gray scale gradient vectors in the x and y directions respectively
H	Threshold value
$k_l a$	Gas to liquid mass transfer coefficient (s^{-1})
η_0, η_1	First order cumulative moments of the gray value histogram
μ	Specific growth rate (hr^{-1})
N	Total number of pixels in image
n_I	Number of pixels in gray level I
P	Perpendicular distance of a line in an image from the origin
p_I	Probability of gray value i
θ	Angle of line p from the x-axis in an image
ω_0, ω_1	Zeroth cumulative moments of the gray value histogram
Vvm	Volumetric volumes per minute (min^{-1})
x	Horizontal distance or axis
y	Vertical distance or axis
z_n	Gray value of a pixel in a filter kernel at position n

1 INTRODUCTION

Bacillus thuringiensis is a spore forming, gram positive, rod shaped bacterium. It is a producer of biological, host specific insecticides and thus has commercial importance.

It has been observed and reported in the literature that *Bacillus thuringiensis* undergoes distinct morphological changes through it's lifecycle. These changes include the production of spores and parasporal insecticidal protein crystals, both of which are observable under phase contrast microscopy (Meadows *et al.*, 1992).

Work done at the research and development division of AECI indicated that the control of a fermentation based on the quantification of bacterial morphology held promise. It was hypothesised that the state of a fermentation could be correlated with the bacterial morphology and from this an advanced control strategy could be developed. In addition, the need for a reliable method of counting bacteria was required. This needed to be consistent between different operators working at different times of the day. Using their present method, different operators seldom achieved similar counts for the same sample.

The thesis thus discusses the different image processing and microscopic methods needed to automatically quantify the morphology of *Bacillus thuringiensis*, including the identification of spores and protein crystals. Automatic counting procedures, developed separately, are also discussed.

The commercially available software package Optimas, version 5.2, was used for most image processing. Macro's which automated the processing steps were developed in the Optimas environment and these are discussed. The possibility of directly analysing the overall shape of a bacterium was also investigated and the application of the Hough transform for lines and circles is discussed. The Hough transform is not supported in the Optimas software package and these procedures were thus written in Visual Basic version 3.0.

The literature concerning *Bacillus thuringiensis,* the use of general image processing procedures and image analysis as applied to microbial systems is reviewed in section 2. This section is directly concerned with establishing what is known about the morphology and growth of *Bacillus thuringiensis*, what image analysis techniques have been applied to other microbial systems and what general image analysis techniques could be of use.

Section 3 discusses the methods employed and their mode of operation. This includes physical methods such as slide preparation as well as the image analysis procedures employed.

Section 4 reports on and discusses the results obtained. As the primary aim of the thesis was the development of automatic and robust image analysis methods, this section is mainly concerned with validating the methods developed. Interesting trends observed in the processing of results are also reported and discussed. These serve to indicate the value of future work using the techniques developed in the thesis.

Conclusions and recommendations follow.

2 LITERATURE REVIEW

2.1 INTRODUCTION

Bacillus thuringiensis is a spore forming, gram positive, rod shaped bacterium. It is aerobic and is motile (Claus and Berkeley, 1994). *B. thuringiensis* can be identified by its production of parasporal crystal bodies (Meadows *et al.,* 1992). These crystal bodies contain one or more polypeptides which are both highly toxic and specific to some insect pest larvae (Weinzerl and Solter, 1997). *B. thuringiensis* is thus an important producer of biological, host specific insecticides.

B. thuringiensis shows a variety of morphological changes during its life cycle. Cells have been observed to vary in length, to develop amorphous intracellular crystals and granules when grown in adverse conditions, and to produce both spores and protein crystals (Arcas *et al.,* 1984). Quantification of the morphological changes that *B. thuringiensis* undergoes during its life cycle could be potentially advantageous for the study of the micro-organism, its response to various environmental conditions and as a proper tool for the control of fermentations.

In this chapter the literature concerning the morphology of *B. thuringiensis* is first reviewed. This is followed by a discussion on the use of image analysis for the quantification of other microbiological systems. Further work in image processing is then discussed. This work is of importance in that it discusses new techniques which apparently have not as yet been applied to the image analysis of microbiological systems, or which need explanation. The chapter is concluded with a discussion of the application of image processing to *B. thuringiensis* and the challenges that this poses.

2.2 THE MORPHOLOGY AND GROWTH OF *BACILLUS THURINGIENSIS*

The genus *Bacillus* is characterised by rod-shaped cells; the formation of endospores which are very resistant to adverse conditions; the presence of a single spore per cell; and the fact that sporulation is not repressed by exposure to air. This genus is generally gram positive, with most species being widely distributed in nature (Claus and Berkeley, 1994). *Bacillus thuringiensis* may be distinguished from the rest of the genus in that it produces a parasporal intracellular crystal inclusion (a δ-endotoxin) that has insecticidal properties (Bulla *et al.*, 1977).

2.2.1 Typical Dimensions and Morphologies of *B. thuringiensis* in the Literature

B. thuringiensis has a width of 1.0 - 1.2 μm and a length of 3 - 5 μm. During sporulation most subspecies do not exhibit swollen sporangia, and the spores produced are ellipsoidal and generally centrally situated in the cell (Claus and Berkeley, 1994). Table 2.1 presents the characteristics of *B. thuringiensis.*

Table 2.1: *Typical properties of* B. thuringiensis, *Claus and Berkely (1994)*

Characteristics	*B. thuringiensis*
Width of rod (μm)	1.0-1.2
Length of rod (μm)	3-5
Spore shape	Ellipsoidal
Spore position	Central
Sporangium swollen	-
Parasporal crystals	d
Capable of Anaerobic growth	
Growth pH	+
Temperature Range	5.7 – 6.8 and above
	10°C – 40°C

Symbol: -, 90% or more of strains do not exhibit characteristic

+, 90% or more of strains exhibit characteristic

d, 11-89% of strains exhibit characteristic.

2.2.2 Stages in Sporulation and the Production of Protein Crystals.

All *Bacillus* species exhibit similar stages in the formation of their endospore. These are:

1. Vegetative cell growth.
2. Preseptation - where the DNA forms an axial filament.
3. Septation - separation of chromosomes resulting in asymmetric cell formation.
4. Engulfment of the forespore, where the membrane of the developing spore becomes completely detached from that of the mother cell and surrounds the spore protoplast.
5. Start of cortex formation.
6. Synthesis of spore coats.
7. Development of refractility, heat resistance and spore maturation.
8. Lysis of the sporangium and liberation of the mature spores.

(Claus and Berkeley, 1994)

The mechanism of protein crystal formation and its dependence on sporulation appears to differ between subspecies of *B. thuringiensis*. According to Bulla *et al.* (1977), the protein crystal for *B. thuringiensis* is formed outside the exosporium (region around the spore) within the bacterial cell during stages 4 to 6 of sporulation. This assertion has been modified by Agaisse and Lereclus (1995), who identified two different gene promoters for the expression of the δ-endotoxin protein, one being sporulation-dependent and the other not. For the subspecies *kurstaki* used in this study, which employs the *cryIA* gene to express the δ-endotoxin protein toxic to *Lepidoptera*, Agaisse and Lereclus found that the expression of this gene and hence the production of the δ-endotoxin protein, was sporulation-dependent. Therefore the *kurstaki* subspecies should exhibit crystal formation during or after spore formation (Agaisse and Lereclus, 1995).

Agaisse and Lereclus (1995) also reported that the two promoters needed for the expression of the *cryIA* gene were used sequentially. These promoters (BtI and BtII) were only active between 2 and 6 hours (BtI) and from 5 hours onwards (BtII) after the end of the exponential phase. As the formation of protein crystals depends on

these promoters it is unlikely that protein crystals are produced within 5 hours of the end of the exponential phase.

B. thuringiensis also produces an exotoxin. This is linked to the exponential growth of *B. thuringiensis* (Holmberg *et al.,* 1980). Tzeng and Young (1995) reported maximum exotoxin production, using the subspecies *darmstadiensis,* at the transition between the exponential and stationary phases. Sporulation occurred 5 hours after the onset of the stationary phase.

The parasporal δ-endotoxin crystals formed by different subspecies of *B. thuringiensis* exhibit different morphologies. Meadows *et al.* (1992) used the presence of these parasporal δ-endotoxin crystals to differentiate *B. thuringiensis* from *Bacillus cereus* and then further classified *B. thuringiensis* subspecies according to the morphology and toxicity of the δ-endotoxin crystal. They reported that the *kurstaki* subspecies has a bipyramidal or near bipyramidal crystal structure, although the presence of differently shaped crystals may be present in a sample. Other crystal morphologies were also identified, but the majority of the isolates (53%) had bipyramidal crystals and these exhibited the highest toxicity.

When the external pH falls to below 5.5, *B. thuringiensis* may produce intracellular granules and amorphous crystals (Arcas *et al.,* 1984). Agaisse and Lereclus (1995) report the onset of a non-dividing semiquiescent period as a response to nutrient stress. They do not state if the cells display a distinguishable morphology while in this stage but do report that it is an alternative pathway to spore formation.

2.2.3 Fermentation and Media Conditions Required for Optimum Growth of *B. thuringiensis* and the Production of Proteins

Arcas *et al.* (1987) performed a series of batch and fed batch experiments in order to determine the effect of nutrient concentrations on the growth and sporulation of *B. thuringiensis* var. *kurstaki.* This was based on their previous work (Arcas *et al.,* 1984) in which they had established the importance of various nutrients for the

production of spores and the δ-endotoxin. In this earlier work they found that the most suitable carbon source was glucose. Ammonium sulphate was shown to be an acceptable nitrogen source but that the amino acid cysteine was also required, ideally from yeast extract. The addition of magnesium, calcium and manganese was found to increased protein yields substantially. They reported spore concentrations of up to 2.1 $\times$ 10^9 spores.ml^{-1}. In their later paper (Arcas *et al.*, 1987) they determined the optimum concentration of glucose, yeast extract, ammonium sulphate and salts required to produce maximum yields of spores and proteins as well as the optimum media composition for biomass yield (Table 2.2). Their fermentations were carried out in a 6 litre fermenter at 30°C, agitated at 500 rpm and with an aeration rate of 0.5 vvm. The maximum spore concentration achieved in this work was 7.36 $\times$ 10^9 spores.ml^{-1}. Unfortunately in both papers Arcas *et al.* (1984, 1987) analysed for spores based on the number of colony forming units (CFU's) produced on a plate. Their bacterial counts were based on biomass. It is therefore not possible to deduce a ratio of percentage sporulation from their work.

Table 2.2: *Optimum medium composition for growth of B. thuringiensis (from Arcas et al., 1987).*

Medium Concentration (× Standard *)	**Lag Period (h)**	**Maximum Biomass concentration (grams dry weight per litre)**	**Specific Growth Rate (μ) (h^{-1})**	**Biomass yield (grams dry weight per gram of O_2 consumed)**	**Spore concentration per ml × 10^9 (As measured by colony forming units)**
1	1.7	4.02	0.77	1.50	1.08
3	1.7	12.53	0.74	1.17	3.22
5	3.5	20.07	0.72	1.01	5.31
7	3.9	26.60	0.71	0.80	7.36
9	5.7	32.46	0.64	0.74	6.92
11	7.2	27.96	0.47	0.56	4.92

Notes : * Standard medium contained: glucose (8 g/l), yeast extract (4 g/l), ammonium sulphate (1g/l), $CaCl_2.2H_2O$ (0.04 g/l) , $MgSO_4.7H_2O$ (1 g/l), $MnSO_4.H_2O$ (0.03 g/l) , KH_2PO_4 (1 g/l)

Although Arcas *et al.* (1987) reported maximum biomass yields at 9 times the standard medium concentration, they found that protein production peaked at 7 times the standard medium concentration. The fermentation media used in this study corresponds closely to the 5 times standard concentration medium in Table 2.2, with the exception that it contains corn steep liquor and lower concentratons of yeast extract and ammonium sulphate (See Section.3.1.2).

Holmberg *et al.* (1980) monitored the growth of bacteria in a batch fermentation of *Bacillus thuringiensis* var. *thuringiensis*. They used a medium based on soya and molasses. Figure 2.1 shows the bacterial concentrations they achieved and the percentage of spores found per bacteria. They counted bacteria through a plate count method and determined percent spores through a dry staining technique.

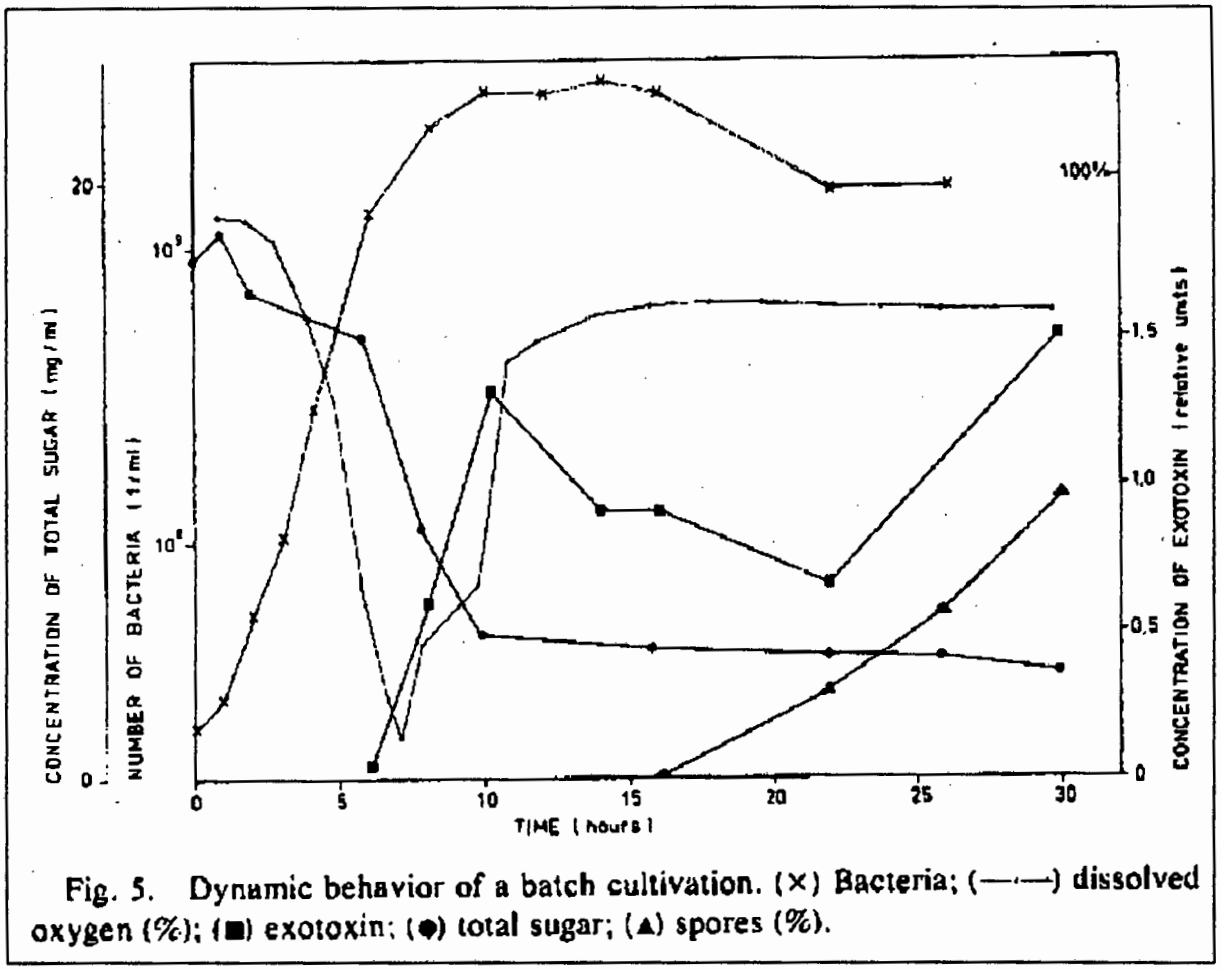


Fig. 5. Dynamic behavior of a batch cultivation. (×) Bacteria; (—·—) dissolved oxygen (%); (■) exotoxin; (●) total sugar; (▲) spores (%).

***Figure 2.1** : Bacterial concentration and % sporulation as reported by Holmberg* et al., *(1980)*

Figure 2.1 clearly shows the onset of sporulation at around 5 hours after the onset of the stationary phase of growth.

2.2.4 Literature studies showing the morphologies of other bacteria

As no work on the morphology of *B. thuringiensis* could be found, it could be useful to discuss studies on the morphology of other bacteria which may give an indication of possible morphological trends and changes in *B. thuringiensis*. A number of studies have been reported in the literature which quantify the changes in morphology, motility and adhesion properties of bacteria at different phases in bacterial growth and under different physicochemical conditions.

James *et al.* (1995) reported changes in the morphology and adhesion properties of *Acinetobacter* under different nutrient conditions. They observed the bacterial population in a flow cell. They reported a change in the cell morphology from rod-shaped cells to cocci by reduction-division when the cells were irrigated with a medium of low nutrient concentration. They also reported that cells in media with low nutrient concentrations remained firmly attached to the glass of the flow cell in which they were observed. In contrast, cells growing in media containing high concentration of nutrients exhibited low adhesion and a tendency to drift and transiently attach to the glass of the flow cell.

Donachie *et al.* (1976) reported that with *E.coli,* cell length increased with increasing growth rate and that the cells divided once they had reached twice the minimum cell length.

Lawrence *et al.* (1992) analysed the behaviour of *Vibrio parahaemolyticus* (*V. parahaemolyticus*) variants under different media viscosities. They showed that *V. parahaemolyticus* changes from having polar to lateral flagella with increasing viscosity. This influences the motility of the cells and hence they are able to remain motile at viscosities of up to 1000 cp when laterally flagellated, whereas they would become immobile at viscosities of only 60 cp when polarly flagellated. These authors also observed that many bacteria are more motile at viscosities of 2 cp than they are in water, although they did not attempt to explain why. This observation indicates that any dilution in sample preparation may influence cell motility.

2.3 THE USE OF IMAGE ANALYSIS IN THE STUDY OF MICROBIAL SYSTEMS

2.3.1 General Applications of Image Analysis for Microbial Systems

The image analysis of microbial systems is a field which has shown rapid growth in the last few years. Microbial systems in which image analysis has been successfully applied include the study of the morphology of single bacteria (for e.g. James *et al.*, 1995; Singh *et al.*, 1990; Sieracki *et al.*, 1989), the automated counting and the morphological study of *Saccharomyces cerevisiae* (Pons *et al.*, 1993; O'Shea and Walsh, 1996; Huls *et al.*, 1992), studies on the morphologies of fungal pellets and mycelia (Cox and Thomas, 1992), studies on plant cells (Kieran *et al.*, 1994) and animal cells (Wu *et al.*, 1995). Despite the diversity in the organisms under investigation, they all have common problems with regards to sampling, calibration of their systems, the acquisition of sufficient images for statistical significance and the choice and possible automation of image analysis techniques. Some workers have also made attempts to quantify cell motility (Phillips *et al.*, 1994).

The overall aim of applying image analysis to study microbial systems includes:

- Understanding the influence of environmental factors on the organisms morphology (James *et al.*, 1995).
- Accurately sizing cells (Sieracki *et al.*, 1989).
- Identifying and quantifying bacterial injury (Singh *et al.*, 1990).
- Automatically counting cells (Costello and Monk, 1985).
- Characterising and classing filamentous micro-organisms (Cox and Thomas, 1992).

It is thus evident that the field holds potential in increasing the understanding of microbial systems. This is true both to increase the scientific knowledge of the organism under investigation and the potential control of industrial processes which may utilise that organism.

2.3.2 Preparation of a Sample for Image Analysis

For offline analysis, the preparation of a sample usually involves the preparation of a slide which then can be observed under a microscope. This could be a flowcell type arrangement (James *et al.*, 1995; Yamashita *et al.*, 1993), a counting chamber (Costello and Monk, 1985) or a normal microscope slide (O'Shea and Walsh, 1996). When using the standard microscope slide, the sample may be pre-treated, usually air dried and stained (Adams and Thomas, 1987), or fixed with a suitable fixative (Singh *et al.*, 1990). Dry and fixed samples are used mainly to classify cellular or mycelial morphology. An advantage of drying or fixing samples is that effective stains may then be used to highlight cellular characteristics of interest. This also assists any subsequent image processing.

For automatic counting and estimation of cell concentration it is obviously important to know the volume of sample taken. A flow cell or counting chamber is thus ideal since both have fixed, known volumes. The use of a wet preparation slide is more contentious for the determination of cell concentration, even when a set volume of liquid is used for the preparation of the slide. For example, O'Shea and Walsh (1996) used 20 µl of sample per slide for their slide preparations. They do not, however, state how they ensured uniform distribution of liquid across the slide or how they overcame the influence of capillary flow. On the other hand, Packer *et al.* (1991) used a 4µl sample per slide and then photographed the entire slide so as to exclude any variability over the coverslip area. Here the entire 4µl sample was analysed. They also analysed the statistical variability between slides in order to quantify the error introduced in the preparation of the slide as well as the error introduced by sample dilution. Their results showed that slide preparation introduced an error of 8.3% (maximum) between 10 slides, while sample dilution introduced an error of 7.3%. This dilution error occurred when older, larger, mycelia were analysed. In the case of younger mycelial samples they found that lower errors occurred (in the order of 4%) as the mycelia were smaller and more dispersed. As *B. thuringiensis* is unicellular, it should have an even lower dilution error, provided clumping does not occur and influence the dispersion of the bacteria in a sample.

At present most image analysing is performed off-line, although some flow cell arrangements have been configured on-line (Treskatis *et al.*, 1996). A novel technique involving a microscope mounted into the port of a bioreactor for the direct visualisation and counting of yeast cells has been described by Suhr *et al.* (1995). Here the number of bacteria in a set volume is determined by calculating their distance from the plane of focus of the microscope (based on how blurred their edges are). This technique could not however be used for more detailed morphological characterisation as most of the cells counted are out of focus or blurred due to movement.

2.3.3 Number of Images Acquired

The acquiring of images from a slide is usually based on a threshold number of organisms counted. This can either be determined by the computer during counting such that once a sufficient number of organisms have been identified no more images are acquired (O'Shea and Walsh, 1996), or a set number of images may be chosen which ensures that a minimum number of organisms are present (Cox and Thomas, 1991). O'Shea and Walsh (1996) used a threshold of 300 cells, while Nielsen and Krabben (1995) took 50 images. The latter ensured that the standard deviation in the average total hyphal length measured was less than 5%. The actual acquiring can be either entirely automatic (Cox and Thomas, 1992), using a motorised microscope stage, or manual. Usually this dictates the method by which images are chosen, either in a grid-like manner across a slide or randomly.

2.3.4 Calibration of the Image Analysis System

It is important to ensure that any dimensions or other measurements extracted by the image analysis system are correct. This requires accurate calibration of the system using images in which particles or grids of known dimensions are seen.

Image analysis systems are calibrated using either beads of known size (Sieracki *et al.*, 1989) or using the grid of a counting chamber (Adams and Thomas, 1987). It is important to ensure that the images obtained are corrected for aspect ratio (i.e. that there vertical and horizontal scales are the same) (Pons *et al.*, 1993), thus ensuring that the orientation of the organism in the image will not influence any measurements taken.

2.3.5 Image Processing Routines

Routines employed in the literature varied widely in the algorithms employed and the degree to which human intervention was required. A routine in which no human intervention was required is termed an automatic routine, one in which only a minor human input was required is termed a semi-automatic routine (Gonzalez and Woods, 1992).

The routines discussed in the literature employ all or some of the following steps in the analysis of an image:

- Preprocessing in which morphological operations which aid in thresholding and thus in the identification of objects of interest are applied. Edge detection filters and noise reduction filters are examples. This step often requires large computational power and is thus often not performed.
- Thresholding in which the image is divided into foreground and background, where the foreground consists of the objects of interest. This is normally based on objects being either brighter or darker than their surroundings and is often the only manual step in an image processing routine.
- Binary processing to separate objects, fill holes, remove noise and smooth edges.
- Application of shape finding algorithms such as the Hough transform to locate characteristic shapes of interest. This step is highly computationally expensive and is seldom applied.

- Extraction of data from the image, either directly from the binarised image or by using the binarised image as a template to the original image.

(Gonzalez and Woods, 1992)

The image processing routine chosen is dependent on the processing power available and the quality of the image to be processed. Some workers simply manually threshold the image and then extract data from the binarised image (James *et al.*, 1995). Others performed binary image processing on the image after thresholding (Cox and Thomas, 1992; Paul *et al.*, 1992). Preprocessing before thresholding and binarisation to highlight edges or other regions of interest is also reported (Yamashita *et al.*, 1993).

Preprocessing involves reducing noise or highlighting regions of interest for later identification based on some criterion such as rate of change of gray level (a characteristic of edges). Reducing noise is usually performed by applying a median or mean filter. The mean filter replaces the gray scale value of a pixel with the average gray scale value of the surrounding n×n environment. This filter was applied by Adams and Thomas (1987) to reduce noise. Its application is, however, contentious in any application where edge detection is required as the mean filter tends to blur the image and distort the edges (Gonzalez and Woods, 1992). Gonzalez and Woods (1992) claim that the non-linear median filter is better as it does not shift edges and is good at removing random, point like noise. When applying this filter, though, highly textured images may loose some detail (Dougherty, 1994).

Besides noise reduction techniques, there are a large number of other pre-processing steps designed to highlight regions of interest. These include subtraction of a background image to correct for non-uniform illumination (Pons *et al.*, 1993) and the application of an edge enhancement filter (Adams and Thomas, 1987).

After preprocessing has been performed the image is normally thresholded and binarised. As the thresholding step normally determines whether the overall routine is either manual, semi-automatic or completely automatic, thresholding is discussed in Section 2.3.6.

If necessary, binary processing can be performed after thresholding. This usually involves applying binary dilations or erosions or combinations of the two (binary openings and closings) to the image. The aim is to separate touching objects (Reichl *et al.,* 1992), remove extraneous data, e.g. dust (Pons *et al.,* 1993), the hole filling of objects identified from their edges (Pons *et al.,* 1993) and the rebuilding of poorly found cells (O'Shea and Walsh, 1996).

Once binarised and processed, objects can be identified and sorted based on shape or dimension criteria. Sometimes further processing is required to separate touching objects. This has been attempted by binary eroding to ultimate erosion markers (O'Shea and Walsh, 1996), outlining and separating through a series of logical arithmetic operations (O'Shea and Walsh, 1996), or the use of concavity in cell junctions (O'Shea and Walsh, 1996). None of these techniques worked on its own when applied by O'Shea and Walsh (1996). They required combinations of all three for the processing of different cells in an image. They were also unable to separate buds from yeast cells using any of the above methods.

Pons *et al.* (1993) identified the number of yeast cells in an unseparated object based on the number of regional maxima in the distance transform of the binary image. They did not attempt to separate the objects. From this it is postulated that a separation method based on the distance transform could be a new and powerful technique for separating touching cells. A separation of touching coffee beans has been successfully applied using the watershed algorithm applied to the inverted image of the distance transform (Beucher and Meyer, 1993).

A different approach to identify objects in an image is to find characteristic shapes or patterns (Leavers 1992a). The Hough transform identifies objects based on their shape. A disadvantage is that it requires large amounts of computer memory and long

processing times, especially for complex objects (more complex than combinations of lines or circles). It does not, however, require continuous or completed edges and can be used to identify occluded objects (Leavers, 1992a). Preprocessing to generate a binary image corresponding to the edges in the original image is normally required before the application of the Hough transform. While attempting to identify yeast cells, Yamashita *et al.* (1993) used a median filter to reduce noise without affecting the edges of the cells. They then applied the Sobel operator to highlight edges before thresholding and binarising. Particle detection was then based on the Hough transform for circles. This was based on the assumption that yeast are almost circular. They conclude that the Hough transform is less computationally expensive than a pattern matching algorithm.

2.3.6 Automated Image Processing

Most work in the literature has been performed in a semi-automatic manner. This implies at least some human intervention and judgement during the processing of images. In some cases this intervention has merely been to exclude cells that have been mistakenly identified or incorrectly classified (Adams and Thomas, 1987). In most cases, however, the thresholding of the image is one of the manual steps (O'Shea and Walsh, 1996; Cox and Thomas, 1992). This could introduce errors as it is based on operator judgement.

The automation of an image processing routine relies on the computer to completely process the image and decide which objects in the image are cells and which are not. This process requires automatic thresholding to identify the cells (with or without preprocessing), either because they are darker or lighter than their surroundings. Thresholding is quite often the only manual step in semi-automatic image processing routines, such as that implemented by Pons *et al.* (1993) . In other cases the thresholding is left at a preset value and not dynamically set, although the image processing routine used is fully automated (Packer and Thomas, 1989). This would obviously rely on the background maintaining a uniform gray level between different images (i.e. each image is illuminated to the same extent and the background therefore appears to be the same shade of gray).

The best method would depend on the computer automatically setting the threshold, either based on a histogram of gray levels in the image or some other information. In setting the threshold based on the gray level histogram it is assumed that the foreground and background each produce a "peak" in the histogram, separated by a "valley" (Otsu, 1979). The histogram for the entire image should then exhibit two peaks, one for the background gray levels and the other for the foreground (more if there are different types of object). This requires a sufficient contrast in intensities between the cells and the background (Wu *et al.*, 1995). Vincente *et al.* (1996) automatically thresholded their images of yeast flocs by applying the method proposed by Otsu (1979) (see Section 2.4.4). They validated their results by plotting the area distribution of floc (i.e. the area attributed to the foreground) against a changing threshold. Regions where this stayed constant corresponded to straight lines on the plot and represented the optimum range in which the automatically determined threshold should lie. They found that their automatically determined threshold did indeed fall in this range. This can be expected as the automatic threshold selection method is based on the zeroth and first order cumulative moments of the histogram.

2.3.7 Processing Times and Statistical Relevance

The processing time is directly linked to the type of data required and this depends on the application. It is therefore not relevant to compare different processing times as these are application and system dependent. Processing times, however, vary from less than a minute per image to well over an hour.

Data extracted for fungi in the literature include mycelial length, branching characteristics and pellet morphologies (including pellet density) (Olsvik *et al.*, 1993). The determination of pellet density was based on the compactness of the pellets. This is defined as the ratio of the area of the pellets with internal holes excluded divided by the total area of the pellet including holes. The quantification of pellet density is potentially beneficial to this study since *B. thuringiensis* should become brighter as it approaches sporulation and protein crystal production and should thus exhibit "holes"

in the cells which could be considered as an indicator as to the physiological state of the cells.

For single cells, cell area, cell length and cell breadth are measured. James *et al.* (1995) defined cell length as the maximum diameter of the *Acinetobacter* under study and the cell breadth as the minimum diameter. Since *Acinetobacter* is either rod shaped or coccoid this, appears to be acceptable. For the measurement of yeasts, which are often assumed to be prolate spheroids, O'Shea and Walsh (1996) used the ferret length and width to characterise the cells. They then used these measurements to calculate the volume of the cells. They noted that this produces errors when the cells have concave curves. The geodesic length and a width based on the distance transform has been used by Pons *et al.* (1993) for measuring yeasts. Here they identified the pixel with the largest minimum distance to the edge of the cell and defined the width as twice this minimum distance.

In some cases the compacity or circularity of an object is measured or used as a criteria in image processing (Pons *et al.*, 1993). This is a dimensionless group based on the ratio of the object's perimeter squared to it's area, i.e.:

$$c = \frac{perimeter^2}{area} \quad or \quad c = \frac{perimeter^2}{4 \cdot \pi \cdot area} \tag{2.1}$$

and is used to describe how close an object approaches a circular shape, where a circle would have a value of either 4 π or 1 in the above equations, respectively. Paul *et al.* (1992) used a circularity method to identify spores of *Penicillum chrysogenum.* This method is of course negated if the image processing method itself is based on shape. Yamashita *et al.* (1993) used the Hough transform to identify yeast cells. As this approximates each yeast cell to a circle, any underlying circularity data is lost.

Once obtained, the accuracy and relevancy of the data needs to be evaluated. This task is not trivial. Semi-automatic methods which rely on operator judgement are subject to bias, while completely automatic methods need to be compared to something in order to quantify the extent of uncertainty in the measurements. This is in addition to the inaccuracies introduced during image acquisition and the inherent loss of data associated with digitising an (analogue) image. Although classifying the

image from a high magnification image as analogue is possibly contentious, the wavelength of light in effect digitises the image, making resolution finite. Wu *et al.* (1995) demonstrate a closeness of fit by overlapping their found cell image outline onto the original image. This of course then relies on the human eye-brain system to assess, with the possible introduction of bias.

Methods which correlate cellular morphology as found by the image analysis system with volume distributions found on other systems have been reported. Volume distributions can be obtained by flow cytometry or electronic particle counters, but these distributions are relative, not absolute (Huls *et al.*, 1991). Huls *et al.* (1991) compared these distributions to distributions obtained through image analysis and found a reasonable agreement.

If the image processing is simply concerned with counting the cells, or classifying them into categories, then the problem of validating the image processing system is simplified. Cell counts may be compared to manual counts in a counting chamber, to optical density or even to dry mass (Nielsen and Krabben, 1995), although dry mass is dependent on cell size and therefore a volume measurement is more likely to be accurate. If a classification system (based on shape) is required then the computers performance can be evaluated manually as the classification is essentially subjective. O'Shea and Walsh (1996) used image analysis to classify yeast cells into 1 of 6 categories, ranging from simple ovoid yeast cells to branched mycelia. They reported an error of 5.6% in a sample. Some of their image processing steps were, however, manual.

Edge effects in images also potentially introduce errors. A cell which extends beyond the image boundary should be excluded as only the portion within the image would be measured and this would influence the overall results by introducing an error. The extent of error is dependent on the relative sizes of the cells and the image. Thus, in a large image with small cells a low overall error is expected. For this reason most studies on the image analysis of single cells in the literature do not appear to correct for biased data as a result of edge effects. In contrast, most work on filamentous micro-organisms use the concept of an active measurement frame (Packer and

Thomas, 1989; Paul *et al.*, 1992). This is a frame within the image with a preset distance from the image edge. The preset distance is based on the largest size of the micro-organisms to be measured. Any micro-organism which lies partially or completely in this frame is then counted or analysed. Any micro-organism entirely outside of this frame is discarded. This accounts for errors in edge effects but does significantly reduce the useful area of an image.

2.3.8 The Quantification of Bacterial Motion.

The motility of micro-organisms during the course of a fermentation can be considered as an indication of the physiological state of the organisms. It is thus relevant to review the work carried out in the area of bacterial motility and its measurement as reported in the literature. Bacteria generally exhibit a "tumble and run" type motility (Figure 2.2) i.e. they exhibit short, smooth runs followed by brief tumbles. The movement is thus random, but the length of the run can be influenced by chemo-, phyto- or aerotaxic gradients (Poole, 1990). Lawrence *et al.* (1992) noted that motility for *Vibrio parahaemolyticus* (*Vp*) varied from "tumble and run" for the polarly flagellated cells to much longer runs (>100s) punctuated by a type of reciprocating motion for laterally flagellated cells.

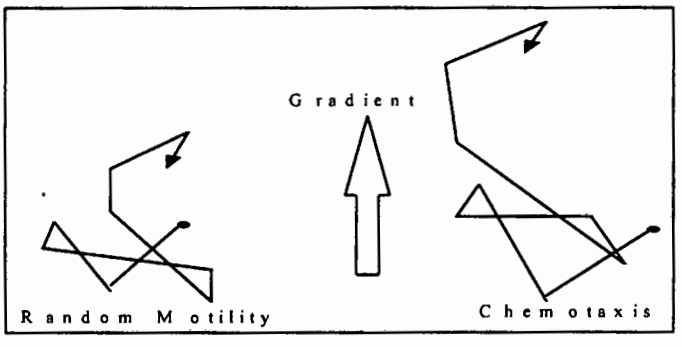


Figure 2.2 : *Bacterial motility in the absence and presence of a chemical gradient (Phillips* et al, *1994).*

The extent of motility can be estimated from video taped recordings. In this way the cellular swimming speed during free runs for the bacterium *Pseudomonas fluorescens*

was measured by Camper *et al.* (1993). Phillips *et al.* (1994) measured the single cell parameters of speed, tumbling probability (how often in a given time period the bacteria tumble) and an index of directional persistence (whether they are less likely to tumble when swimming in a particular direction) by using both microscopic and macroscopic techniques. Their microscopic techniques involved following individual cells under a microscope while recording them. To ensure that the length of run distance was accurate they had to ensure that the entire run was in focus. To do this they set their focal plane to be midway between the top and bottom of the flow cell so as to minimise boundary effects.

The methods described above are heavily dependent on manual operation and control. No automatic determination of cell motility has been found in the literature.

2.4 IMAGE ANALYSIS AS A TECHNIQUE FOR THE STUDY OF DIGITAL IMAGES: USEFUL CONCEPTS

Image analysis is the analysis of scene data for the extraction of useful information about objects or regions of interest. For the purposes of this thesis and review, image analysis refers both to the processing of images to highlight interesting features and to the extraction of data from processed images. Gonzalez and Woods (1992) refer to image analysis as digital image processing. They state that the main steps in any image analysis application are as follows:

- Image acquisition and image storage.
- Preprocessing of the image. This is intended to increase the chances of success of any further steps. Preprocessing typically involves noise reduction, contrast enhancement and possibly the isolation of regions based on texture.
- Segmentation of the image (thresholding) - this partitions the image into it's constituent parts or objects. It is often based on luminance values.
- Feature selection. This involves extracting features that result in some quantitative information of interest, or features that are basic for differentiating one class of object from another. For the purposes of this review this will include binary processing and advanced methods for processing objects, including the use of the watershed, distance and Hough transforms.
- Recognition and interpretation - here recognition involves assigning a label to an object, while interpretation involves assigning meaning to an association of recognised objects.

Each of these steps needs to be optimised for a given system to ensure that the data extracted from images is representative and accurate.

2.4.1 Image Acquisition and Storage.

Image acquisition requires an imaging sensor and the capability to digitise the image produced by the imaging sensor (Gonzalez and Woods, 1992). In this study the imaging sensor is an RGB digital camera, while the digitisation of the image is performed by a frame grabber board.

Image storage refers to the storage and archiving of images for later retrieval. It must accommodate the dual demands of maintaining high fidelity of the original digital image and the requirement that stored images should take up as little memory as possible (Wallace, 1992).

Images are stored either in a compressed or uncompressed form. Here image compression addresses the problem of reducing the amount of data required to represent a digital image. It is based on the assumption that images contain redundant data which can be removed (Gonzalez and Woods, 1992). Compression is either through lossy compression or lossless compression. Lossy compression refers to compression which is non-invertable, whereas lossless compression is invertable (Dougherty, 1994). It is therefore possible to completely reconstruct an image compressed through lossless compression. The compression ratios achievable in lossless compression range from 2:1 to 10:1. Lossy compressions range from compression ratios of 10:1 upwards (Gonzalez and Woods, 1992).

The JPEG image compression method used for the storage of images in this thesis is a lossy compression method and can therefore introduce errors in the original image. A discussion on the acceptability of this method can be found in the sections 3.3.3 & 4.1.2. JPEG stands for Joint Photographic Experts Group (Wallace, 1992). It is based on the Discrete Cosine Transform (DCT). The JPEG scheme uses a combination of spatial-domain and frequency-domain coding. The image is first divided into 8x8 blocks, each of which is transformed into the frequency domain using the DCT. Once transformed, the signal tends to concentrate in the lower spatial frequencies, enabling high-frequency components to be discarded without substantially affecting the appearance of the image. The main source of information loss is in the quantisation

of the DCT coefficients. These are then ordered and entropy coded, using either Huffman or arithmetic coding (Shneier and Abdel-Mottaleb, 1996).

2.4.2 Noise Reduction Techniques

Many images contain extraneous data present from the acquisition of the image (noise). The removal of this noise is best done during image acquisition, through a technique such as frame or line averaging. This is not always possible and the application of a noise reduction filter is then required (Gonzalez and Woods, 1992).

A common and useful noise reduction filter is the median filter. The median filter is a non-linear filter introduced by Tukey (1974) as a method of smoothing economic time series (Astola and Dougherty, 1994). Median filters are efficient at for removing noise from an image while preserving edges (Dougherty, 1994; Gonzalez and Woods, 1992). They are therefore superior to the average (mean) filter for noise removal, especially in applications where edge detection is required. The median filter works by replacing the centre pixel in an n×n neighbourhood by the median value of the gray scale values of the surrounding pixels, whereas the average filter works by replacing the centre pixel in the n×n neighbourhood by the average value of the gray scale values of the surrounding pixels. The neighbourhood is usually 3×3 or 5×5. Noble (1996) states, however, that even better noise reduction filters which accurately localise edges are the clos-opening and the open-closing morphological filters. Note that JPEG compression before processing automatically eliminates most impulse type noise as the noise forms part of the high frequency components in the DCT of the image which are discarded in JPEG compression.

2.4.3 Contrast Enhancement

The enhancement of contrast in digital images has two functions: either to improve the image for human viewing or to enhance details for further computer processing (Gonzalez and Woods, 1992).

Edge enhancement as a contrast enhancement technique seems to have the widest application and most use. It is based on the rate of change of gray scale luminance value at a point (Gonzalez and Woods, 1992; Rosenfeld, 1989). A common edge enhancement technique is the application of the Sobel operator, a filter which replaces the gray scale value of a pixel with the magnitude of the gradient vector at that pixel (Yamashita *et al.*, 1993). The magnitude of the gradient vector is:

$$g(x,y) = \sqrt{G_x^2 + G_y^2} \tag{2.2}$$

where

$$G_x = (z_7 + 2 \cdot z_8 + z_9) - (z_1 + 2 \cdot z_2 + z_3) \tag{2.3}$$

and

$$G_y = (z_3 + 2 \cdot z_6 + z_9) - (z_1 + 2 \cdot z_4 + z_7) \tag{2.4}$$

where z_1 is the gray value of the pixel in the top left, z_2 the pixel in the top middle, z_3 the pixel in the top right, z_7 bottom left, etc. of a 3×3 mask.

Other edge enhancement filters include the SharpenHigh, SharpenMed and SharpenLow filters as implemented by the Optimas corporation (Optimas, 1994a). These are strong, medium or low edge enhancement filters, respectively, which replace the central pixel in a 5×5 or 3×3 neighbourhood with a value based on the neighbourhood gray values (see Section 3.5.2 and Appendix G) (Opimas, 1994b).

2.4.4 Automatic Thresholding Techniques

The method proposed by Otsu (1979) was employed in this study, with slight variation as implemented by Optimas corporation in their software. Otsu's method is

based on the assumption that well thresholded classes (objects and background) would be separated in gray levels and conversely that a threshold giving a best separation of classes in gray levels would be the best threshold. The method consists of maximising the separability of classes of gray values (Vicente *et al.*, 1995). This involves the use of the zeroth- and first- order cumulative moments of the gray value histogram. The optimum can be found by maximising:

$$\omega_0\omega_1(\eta_1 - \eta_0)^2 \tag{2.5}$$

where :

$$\omega_0 = \sum_{i=1}^{h} p_i \text{ and } \omega_1 = \sum_{i=h+1}^{255} p_i \tag{2.6}$$

and

$$\eta_0 = \frac{\sum_{i=1}^{h} i \cdot p_i}{\omega_o} \text{ and } \eta_1 = \frac{\sum_{i=h+1}^{255} i \cdot p_i}{\omega_1} \tag{2.7}$$

in which ω_0 and ω_1 are the zeroth- and η_0 and η_1 the first-order cumulative moments of the gray value histogram, respectively (Otsu, 1979). In equations (4), (5) and (6) i is the gray value, h is the threshold and p_i is the probability of gray value i defined by:

$$p_i = \frac{n_i}{N} \tag{2.8}$$

where n_i is the number of pixels with gray value i and N the total number of pixels. The value of h which maximises equation (4) is then the optimum value for a single threshold. Although the Optimas auto threshold function does not explicitly cite any references, tests (Appendix H) have shown that it works in the same way as the method proposed by Otsu (1979). It provides the extra ability of being able to force the objective function (Equation (2.5)) to include or exclude a minimum area in the threshold or to ignore ranges of low or high gray values (Optimas, 1994a).

Once thresholded an image is binarised into foreground (normally white) and background (normally black). Further binary and other processing are then performed, if required, before measurements are extracted. These measurements are either performed on the binary image or on the original image, using the binary image as a mask to identify the objects or regions of interest.

2.4.5 Binary Image Processing

Binary image processing usually involves binary erosions and dilations, or combinations of the two (openings and closings). Hole filling, skeletonisation and outline operations are also common binary images operations (Gonzalez and Woods, 1992; Optimas, 1994a).

A binary opening is an erosion followed by a dilation. It generally smooths the contour of an image, breaks narrow isthmuses and eliminates thin protrusions. A binary closing is a dilation followed by an erosion. It also smooths sections of contours but, as opposed to the opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour (Gonzalez and Woods, 1992). Definitions for other binary operations may be found in the glossary.

2.4.6 The Watershed Algorithm

The watershed algorithm is an advanced technique in the general category of mathematical morphology. It can be used effectively as a method of segmenting gray scale images but has also been applied to the inverted distance transform of binary images. The watershed algorithm may appear similar to the region growing method for segmentation but actually leads to a general methodology for segmenting diverse images (Beucher and Meyer, 1992). In the past the usefulness of the watershed algorithm was limited by the excessively slow processing times and large memory requirements. This has been mostly solved by the introduction of more efficient and robust algorithms (Vincent and Soille, 1991; Bleau *et al.*, 1992). More recently, faster processing times and even better algorithms have been introduced (Moga and Gabbouj 1997; Najman and Schmitt 1996). These have not as yet been incorporated into commercially available software.

It is useful to think of a gray scale image as a contour map, where gray level represents height. This clarifies the notion of minima and maxima in the image as

minima correspond to relatively darker regions and maxima to relatively lighter regions. The watershed algorithm can then be seen to identify the geodesic influence zones (watershed areas) of regional minima by either using a "flooding" strategy or alternatively by finding the zone around the regional minima in which a drop falling will flow to that minima. In the first approach the image is progressively filled from the regional minima, producing "lakes". Where two lakes meet a "dam" is constructed. This corresponds to the watershed between the two regional minima and produces a tessellated image. The second approach identifies pixels as being within an influence zone if a drop falling on that pixel would flow to that zone's associated minima. The lines separating these zones are the watersheds. (Vincent and Soille, 1991)

In practice a watershed algorithm must separate regions in such a way that a few important criteria are met. These are as follows:

- The watersheds (Bleau *et al.* (1992) term these the water partings) must be as thin as possible.
- Distinct watershed areas must be separated by leak-proof watersheds. This dictates the connectedness of the pixels in the watersheds (e.g. if they are 4- or 8-connected).
- The watershed must be located on the outer crest lines of the watershed area. This is the set of the highest boundary pixels having a shortest descending path to the watershed area. This defines where a watershed would lie on a plateau.
- The set of pixels which do not belong to the watershed zone of separation should be homotopic to the actual watershed areas. This means that the number of connected components in the entire image (i.e. areas) must equal the number of watershed areas and that each of these areas should be included in a single and distinct watershed.

(Bleau *et al.*, 1992)

Often the application of a watershed algorithm to an image results in over-segmentation. This occurs because many tiny, shallow watershed areas are found instead of only a few large and deep watershed areas (Bleau *et al.*, 1992).

A number of different methods have been developed to overcome this problem. These include the selective marking of basins in which flooding is to occur (i.e. ignoring certain minima through preprocessing); and processing after application of the watershed algorithm to combine adjacent watershed areas where the original gray level step between the areas is of a low value (i.e. the difference between the minima of the areas and the crest line separating them) (Beucher and Meyer, 1992; Najman and Schmitt, 1996).

In the application of the latter algorithm, Bleau *et al.* (1992) introduced the concept of a 'lake transform' in which the inside of a watershed area is replaced by the minimal altitude of its surrounding watershed. The lake transform is selectively applied to watershed areas which are either below a certain volume or which have a very large surface area to volume ratio (i.e. very small areas or very shallow, wide areas). This effectively eliminates these areas and merges them with adjacent areas. The watershed pre-flood option in the Optimas image analysis software is based on this method.

2.4.7 Separating Touching Areas in Binary Images using the Watershed Separation Technique

For binary images the inverted distance transform is first applied before watershed segmentation. This technique has been successfully applied to the separation of overlapping coffee grains (Beucher and Meyer, 1992). The distance transform has been applied to yeast cells to identify single cells, budding cells and clusters of cells (Pons *et al.*, 1993). No attempt was made, however, to separate these cells.

The distance transform computes the distance between a pixel and the nearest feature pixel and replaces the gray value of the pixel with this distance. In the case of binary images this may, for instance, be the distance between a white pixel and the nearest black pixel. The computation of distance in an image is essentially a global computation and is thus computationally expensive. To overcome this a variety of local approximations of distance based on a small neighbourhood have been

developed (Borgefors, 1986). The algorithm which gives the smallest deviation from the true Euclidean distance, while producing a distance transform of integer valued pixels, is known as the chamfer 5-7-11 algorithm (Section 3.5.6.2). This gives a maximum error of 2.02 % (Borgefors, 1986).

2.4.8 Shape Recognition - the Radon Transform

The most natural method of object detection involves a two step process: a pre-attentive and an attentive stage. It is proposed that this is based on the method employed by the human visual system (Leavers, 1994). The pre-attentive stage involves extracting features from the image, often edge detail. This can then be used by the attentive stage, where shapes or the relative orientation of objects can be used to identify those objects.

I hypothesised that this might be the optimum method of recognising bacteria in an image. The detection of edges can be easily performed by applying an edge detection filter and then binarising the image. A shape recognition algorithm would then be required, ideally to recognise the bacteria as a complete mathematically describable shape, but alternatively as a combination of parallel lines and circles. The Hough transform and the probabalistic Hough transform are the most widely known and used shape detection methods. They form a special case of the Radon transform (Leavers, 1992b). The Radon transform requires only two dimensional transform spaces (the Hough transform requires more than two dimensional transform spaces for any shape primitive more complex than a line) and requires no prior knowledge of the shape under detection. The only parameter is the degree of resolution required since this dictates the size of the transform space.

The transform works by transforming each foreground pixel x_i , y_i in the image into a curve of the form:

$$p_j = x_i \cdot \cos\theta_j + y_i \cdot \sin\theta_j \qquad (2.9)$$

in the transform space (p,θ). The θ values are incremented and the p values calculated. Each point on the cosine curve defined in equation (8) is given a value of

1. Where curves intersect, the point of intersection will have a value equal to the number of intersecting curves. Once the transform has been constructed from all of the foreground points in the original image, the points in the transform space with the highest values (local maxima) correspond to straight lines in the original image. These lines can then be drawn from the p and θ values (Figure 2.3). Resolution is determined by the size of the θ incrementation.

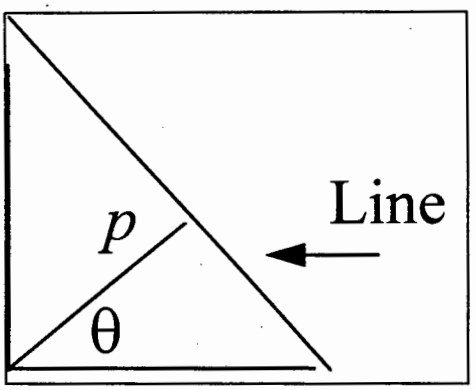


***Figure 2.3**: Graphical representation of the Radon transform*

From: Leavers (1992b)

For arcs of a conic in the image space, the maxima in the transform space occur as constituent points of a sinusoid. Leavers (1992b) then treated the approximately linear sections of these sinusoids as straight line segments and performed the radon transform on this first transform space. The second transform space then contains isolated maxima which correspond to arcs of conic sections in the original space. In this way only two dimensional transform spaces are needed, reducing required memory and processing times drastically when compared to the Hough transform. For a demonstration please consult the visual basic program on the attached CD-Rom.

2.4.9 Analysis of Processed Images - Measurement Extraction

Once an image has been processed to the extent where computer recognition and measurement is possible, the desired measurements are extracted. Sometimes processing itself extracts certain measurements, for example in the processing of lines with the Radon transform the equation describing the line is automatically inferred

from the θ and p values. In other cases the desired measurements still need to be extracted, either using the processed image, or using the processed image as a marker on the original image (this would be required if the gray values in the original image within an object were required). Measurements extracted are then operator determined. Area is often simply the number of pixels in the object times the calibrated area per object. This can introduce errors through the discretisation of the edge of the object (Gonzalez and Woods, 1992), either by estimating the edge to be located on the centre of the pixels which are part of the object but on it's edge, or estimating it to be on the pixels which are not part of the object but are immediately outside of it. In the former case the area can be underestimated and in the latter overestimated. Similar problems exist for determining the length, breadth and other dimensions of an object.

2.5 IMAGE ANALYSIS FOR THE STUDY OF *BACILLUS THURINGIENSIS* : REQUIREMENTS

A single automatic algorithm which can process *B. thuringiensis* throughout its growth would need to do the following:

- Analyse for the number of cells in an image. This should be accurate enough to extract useful data for area, etc. As the identification of spores and crystals during the latter stages is also required, the algorithm should not be disturbed by the halos formed around the cell when viewed under phase contrast.
- Locate and identify spores and analyse for the number of spores in an image.
- Locate proteins and differentiate these from other bright regions in the image. This will probably require knowledge of their inclusion in cells.
- Separate joined cells. This will probably require the application of the watershed algorithm applied to the distance transform of a binary image. Binary openings or ultimate erosions could also possibly work
- Handle artifacts in the image such as out of focus cells, large three dimensional clumps which are undifferentiable and any non-cellular artifacts.

Evidence from the literature indicates that there are many ways to approach a particular image processing application. The image analysis of *B. thuringiensis*, although multifaceted, would probably yield to a combination of the techniques discussed.

3 MATERIALS AND METHODS

The image analysis method chosen and results obtained are dependent on the type and quality of images and thus on the microscopic and fermentation methods employed. This section discusses all methods developed and used, both with regard to image analysis and to the microscopic and fermentation methods. As the primary aim of the thesis is to develop a robust method for the image analysis of *Bt*, this section is dealt with in detail. A validation of the methods is discussed under results (Section 4.1).

3.1 MICROBIAL METHODS

Experiments were conducted using both shake flasks and a bioreactor. Since the focus of this study was to develop and validate a method, most experiments were performed using shake flask cultures. The bioreactor provided a means of validating cell counts and also provided an indication of changes in cellular characteristics due to a different shear and oxygen transfer environment.

3.1.1 Culture Maintenance

Bacillus thuringiensis sp *kurstaki* (a gift from AECI research laboratories Johannesburg, South Africa) was used for all experiments and was stored in a 50% nutrient media / 50% Glycerol solution at –70°C (Sambrook *et al.*, 1989). This was for the long term storage of the culture.

Stab cultures of *Bacillus thuringiensis* were also used for storage and maintenance of the bacteria. These were made by filling mcartney bottles to about 2/3 of their volume with nutrient agar (1gram agar and 1.3 grams nutrient broth in distilled water) and autoclaving at 120°C for 20 minutes. Once the bottles have cooled and the agar has solidified, a thin

sterilised wire with a scraping of *Bacillus thuringiensis* on it's tip is inserted deep into the agar. This is then withdrawn, the lid replaced and the mcartney bottle is incubated at 30°C for 36 hours. This is then stored in a dark place at room temperature. The stab culture should remain viable for a couple of months.

Fresh nutrient agar plates were prepared every 2 weeks for short term storage of *Bacillus thuringiensis*. This is done by taking a sample of fermentation broth, prepared from a stab culture, and wiping a drop of this liquid across the nutrient agar plate. After 24-36 hours incubation at 30°C the plates are stored in a refrigerator and remain viable for 2 weeks.

3.1.2 Media Preparation

Two different media were used in the experiments. The fermentation media consisted of: glucose (40 g/1), yeast extract (20 g/l), ammonium sulphate (5 g/l), $CaCl_2.2H_2O$ (0.2 g/l) , $MgSO_4.7H_2O$ (5 g/l), $MnSO_4.H_2O$ (0.15 g/l) and KH_2PO_4 (5 g/l).

The solutions of glucose, yeast extract, ammonium sulphate and the other salts were prepared separately, neutralised with 5M sodium hydroxide and autoclaved separately at 121°C for 20 minutes (Arcas *et al.*, 1987). They were then mixed aseptically in the correct proportions to produce the fermentation media.

For shake flask cultures, nutrient broth and fermentation media was used. No neutralisation of the broth was performed as it was found that the un-neutralised broth was adequate as a media in which sporulation and protein production occurred. Initial pH of the broth was 7.4 ± 0.2 and ending pH (after cell lysis) was 8.8 ± 0.2.

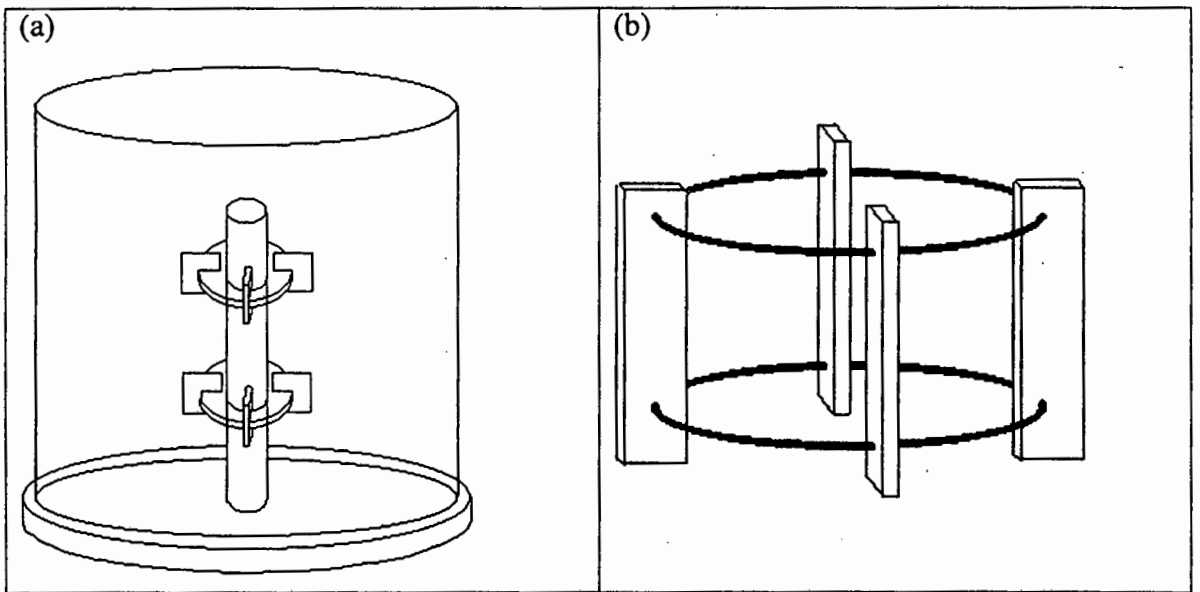
The fermentation media was used in all bioreactor experiments and some shake flask trials. Nutrient broth was used only in shake flask trials. Nutrient broth ensured both sporulation and protein production in the shake flasks whereas the use of shake flasks

with fermentation media did not cause sporulation or protein production, possibly as the pH dropped too low (Arcas *et al.*, 1984).

3.1.3 Bioreactor Geometry and Operating Parameters

The Bioreactor employed for fermentation trials was a Chemap 7 litre batch reactor fitted with pH and temperature control. A working volume of 5.5 litres was used, with an air flow rate of 1.36 vvm and an agitation rate of 900 rpm. High agitation and aeration rates were used to prevent oxygen limitation. Dissolved oxygen was not measured in these experiments but Dabee (1996) reported a k_la value of 0.08 s^{-1} at 800 rpm and 1vvm air flow rate in this same bioreactor.

The impeller consisted of 2 co-axially mounted 4-bladed Rushton turbines on a central shaft, spaced 10 cm apart. The reactor was baffled. Temperature was controlled at 30 °C ± 1 °C. The pH was controlled at pH 7.0 through the addition of 5M NaOH. The quantity of base added to the bioreactor and the pH were logged automatically.



(a) Bioreactor layout with Rushton Turbines.

(b) Baffle arrangement (Baffles situated around central shaft).

Figure 3.1: *Layout of bioreactor and baffles*

Table 3.1: *Dimensions of bioreactor*

Bioreactor Height	300 mm
Bioreactor Diameter	184 mm I.D.
Impeller Diameter	80 mm tip – tip

Bioreactor innoculation was through a sterile port. 500 ml of a stationary phase innoculum was innoculated into the 5 litres of the bioreactor. The innoculum was incubated in a shaker incubator at 30°C. It was in turn cultured from 100 ml of 23 hour old pre-innoculum cultured from a nutrient agar plate scraping. For a complete description of the bioreactor operation, including sterilisation, please refer to Appendix A.

3.1.4 Analyses Performed on Bioreactor Samples

In addition to manual and automatic cell counts, biomass concentration as dry weight was determined. The method used was as follows:

10 ml of sample from the bioreactor was filtered through a pre-dried, pre-weighed Whatman 0.45 μm filter (millipore). The filters were then and dried at 80°C for 48 hours before being re-weighed. This was performed in triplicate for each sample taken.

3.1.5 Shake Flask Geometry and Operating Parameters

Three different shake flasks were used: a 500 ml shake flask, a 1 litre shake flask and a 2 litre shake flask. Since 100 ml of media was put into each shake flask, these served to provide varying levels of oxygen transfer to the culture. All flasks were agitated in a reciprocal shaker-incubator at 100 rpm and 30 °C. The dimensions of the flasks are shown in Table 3.2.

Table 3.2: *Geometry of Shakeflasks Used*

	500 ml	1 litre	2 litre
Height	17.7 cm	21.9 cm	15.0 cm
Width at base	6.8 cm	9.4 cm	19.8 cm
Width at widest point	10.3 cm	12.7 cm	19.8 cm
Surface area of fluid (100ml)	83.3 cm^2	118.8 cm^2	307.9 cm^2

Innoculation of all shake flasks was from 5 ml of 23 hour old innoculum. The innoculum was taken from a nutrient agar scraping and was incubated in a shaker incubator at 30°C.

3.2 MICROSCOPY

The microscope used was an Olympus BX-40. The microscope was fitted with bright field, dark field and phase contrast condensors and 10×, 20×, 40× and 100× objective lenses. Eyepiece magnification was 10×. The RGB Panasonic digital camera was coupled to the microscope through a camera tube.

Three different slides were used for cell mounts. These were a normal microscope slide, a Thoma counting chamber (Weber Scientific) and agar mounted slides. The cell counts were performed by observing cells in the Thoma counting chamber at 1000x phase contrast for manual counting and at 200x dark field for automatic cell counting. Bacterial morphology was quantified by acquiring images from agar mounted slides at 1000x magnification using phase contrast. No images were taken of bacteria on normal microscope slides as the bacteria did not all lie in the same focal plane.

3.2.1 Sample Preparation

In all cases, sample preparation was initiated by diluting 1ml of sample in 9 ml 0.9% saline solution. Samples were agitated briefly with a Scientific Instruments Vortex Genie-2 at a setting of 8 (maximum) to prevent clumping without damage to the cells. It was unnecessary to dilute the samples even at high cell concentrations since the cells were easily differentiable in the counting chamber.

3.2.2 Use of the Thoma Counting Chamber

The Thoma counting chamber (Figure 3.2) has a depth of 20 μm and a grid spacing of 0.2 mm (major grid lines). The known grid spacing was used to calibrate the image analysis

system. Counting of cells in an area corresponding to a fixed liquid volume allowed the determination of cell concentration.

Slide preparation for the Thoma counting chamber involved placing 10 µl of the diluted sample between a thin cover slip placed half-way across the circular region of the counting chamber and the counting chamber itself. The coverslip is then slid across so that it covers the circular region of the counting chamber. The use of a thin coverslip is necessary even though it is more likely to warp and thus distort the volume as thicker coverslips prevent focussing at a 1000× (the objective focusses within the coverslip). The use of a 1000× magnification is necessary as the identification of joined cells is only possible at this magnification.

Manual cell counts were performed on the corner squares of the counting chamber grid and four squares with a total volume of 3.2×10^{-3} µl were counted (Figure 3.2). Two focal layers were evident as most cells became attached either to the glass of the counting chamber or the coverslip. Both focal layers were counted. Typically between 100 and 600 cells were counted in all four large squares.

Automatic cell counting was performed on the same slide preparation as for the 1000× manual counts. Images were acquired from the region outside of the counting grid. Initially 30 images at 1000× magnification were acquired, so that data on cell counts and bacterial morphology could be obtained from the same images. When this was found to be inaccurate, 5 images for counting at 200× dark field were systematically acquired (see Figure 3.2 – the blue squares). The dark field was actually the cells viewed through the phase 3 phase ring using the 20× objective (which normally uses a phase 1 phase ring). The net effect is a dark field image. The five images obtained have a total area of 6.1×10^5 µm^2 and a corresponding volume of 1.23×10^{-2} µl. Typically between 100 and 4000 cells were within the area covered by the 5 images.

Figure 3.2 shows the layout of the Thoma counting chamber (grid shown is the larger grid lines, smaller lines not visible). The green coloured grid boxes are the areas manually counted. Each is 200μm × 200μm. The light blue areas are the areas photographed for automatic counting (not shown to scale). The circular region, including the grid, is 20μm lower than the rest of the slide. The dotted region approximates where a coverslip would lie.

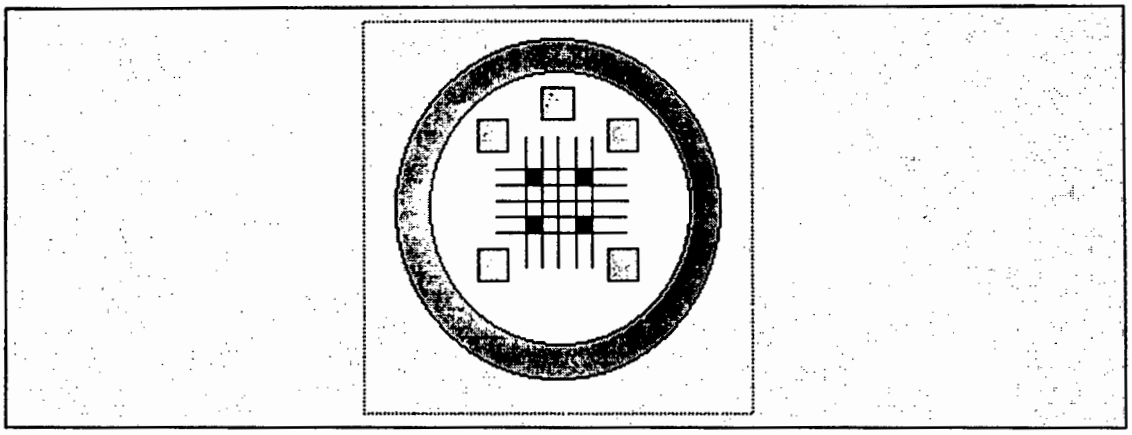


Figure 3.2: The Thoma counting chamber

3.2.3 Agar Slides for Morphology Measurements

Ensuring that the cells are completely in focus proved to be difficult. At certain stages of their growth, notably in late log phase when exotoxin production peaks, the cells tend to attach to the microscope slide and the entire cell is in focus. During other stages cells may be found in other planes of focus. As images containing out of focus cells, or parts of cells, distort the results obtained from the analysis system, a method was required that ensured that all the cells remained in focus. Any method applied should not change the cell volume or morphology, as can be the case with air drying.

A method in which a thin layer of agar is cast on a glass slide, briefly described in Bergey's Manual (Sneath, 1986), was adopted and modified. Twelve slides were arranged in a flat tupperware container and 100 ml of water agar (1% w/v bacteriological agar in distilled water) was poured into the container. This forms a layer of smooth agar over the slides, with the depth of agar above each slide of about 0.5 – 1 mm. The container is closed and stored in the refrigerator to prevent desiccation of the agar. When a slide is needed it is cut out of the container. A 10 µl aliquot of the diluted sample is then placed on the agar layer on top of the slide and a cover slip is placed over this. The slide is then left for a period of about 5 minutes. In this time the sample is slowly absorbed into the agar, leaving the cells flat against the agar. Immediately before observation of the sample, the coverslip is pushed down gently to remove any remaining liquid. The slide is then observed at 1000× magnification using phase contrast illumination. It is necessary to verify the height of the microscope's condenser to ensure that the depth of the agar does not influence the apparent position of the phase ring and thus the quality of the phase contrast image.

3.3 IMAGE ACQUISITION AND STORAGE

The acquisition and storage of images needs to fulfill the dual purposes of acquiring sufficient images to obtain meaningful statistical results while minimising both the time to process and the storage requirements for the images. This section discusses the computer system used and the memory constraints of using the given system as well as the image acquisition and storage methods chosen and how these are necessitated by the computer system. Hardware solutions and their implementation are also discussed.

3.3.1 The Computer System

An Intel pentium-133 with 16 Mb Ram was used to process all images. Images were acquired using a Panasonic RGB digital camera which was fed into an IC-PCI-4× frame grabber. Image storage made use of the 1.2 Gb hard disk (about 800 Mb available for storage). This necessarily limited either the number or quality of the images stored on the hard disk. As an example, with 30 images per sample and 8 samples per experiment, high quality RGB TIFF images (at 1.35 Mb each) would require 324 Mb per experiment. If the images were stored in gray scale this would still require 108 Mb per fermentation. These memory requirements were prohibitive and an alternative method was sought. This is discussed in the section under image storage.

3.3.2 Image Acquisition

Images were acquired in colour to reduce noise in the acquisition process. It is postulated that in gray scale acquisition only one of the three (R, G or B) channels is acquired. In contrast, acquiring in colour and then converting to gray scale combines all three RGB signals, thus averaging out any noise.

Images were acquired directly by the Optimas version 5.2 software package from the frame grabber (i.e. they resided in memory on the frame grabber). The resultant frame grabber dependency of the images meant that only 8-bit gray scale or 24-bit colour images were storable.

The images were acquired in a semi-automatic manner. This had benefits of being quicker and less likely to produce operator bias. A customised macro (Appendix B) was written to acquire images every six seconds. A tone indicates to the operator when to move the microscope stage and focus on a different area of the slide before another tone warns him that acquisition is about to occur. The six seconds was found through experience to be sufficient time to move and focus the slide, while reducing the possibility that an interesting or "dense" region of the slide would be purposely chosen. The macro saved the images with file names such that they could be automatically processed.

Five images of 200× magnification were acquired from the counting chamber for cell counting. The choice of area for each image was systematic, as shown in Figure 3.2. For the earlier fermentation samples, in which 30 images were taken both for morphology measurements and counting, the choice of area on the counting chamber was random.

Thirty images for cellular morphology determinations were acquired. The choice of this number is discussed in Section 4.1.3. These images were not randomly chosen in the sense that any area of the agar was photographed. It was ensured that in the taking of each image that at least one cell was within the image border. The choice of which cells were chosen was random (i.e. an attempt was made to ensure that cells were not specifically chosen based on unusual features or any other characteristic).

3.3.3 Image Compression

As the computer system was being used by other users and all images acquired were stored on the hard disk, image compression was required. The JPEG image compression method was chosen, with a compression factor of 70. Figure 3.3 shows the method applied to an example image (See also attached CD). The JPEG compression scheme is a lossy compression system and was designed to compress images while maintaining image quality for human viewing (Gonzalez and Woods, 1993). It is thus not an ideal system for storing images for later image processing. Compression is rated from a JPEG factor of 0 (maximum compression, minimum quality) to 100 (minimum compression, maximum quality).

a) Region of RGB TIFF file magnified 4 times (in X and Y directions) For area of 768 × 572 pixels; file size = 1.25 Mb	b) JPEG compression of same region at JPEG factor 50. For area of 768 × 572 pixels; file size = 17.1 kb	c) JPEG compression of same region at JPEG factor 70. For area of 768 × 572 pixels; file size = 21.6 kb.

Figure 3.3 : *The influence of the JPEG compression factor*

Figure 3.3 shows the appearance of "blocking", caused by the sectioning of the image into 8×8 blocks in the compression system. The images were not, however, significantly

degraded with regards to the extraction of useful data (Section 4.1.2). The JPEG 70 factor was chosen as optimum for the given system.

The major drawback introduced by JPEG compression was the appearance of edges in an otherwise uniform background. This occurred as a result of the blocking. These edges were sometimes highlighted during image processing and could ultimately cause the identification of false cells. To overcome this a checking procedure on identified cells was introduced. This is based on the uniformity of the gray scale values of the identified object (See Figure 3.5 and Appendix C).

3.3.4 Number of Images per Sample

The number of images chosen per sample was 30 in the case of the morphology measurements and 5 in the case of the automatic cell counting methods. These 30 images per sample gave between 118 and 348 cells in total for the full scale fermentation (excluding the last sample where large scale cell lysis had occurred). This was sufficient to give a maximum error of ± 0.13 microns for the calculated mean cell length and ± 0.04 micron for the calculated mean cell width. Pixel dimensions at this magnification are about 0.1 micron. The maximum error in the mean is therefore adequate given the maximum accuracy's obtainable (i.e. to within 1 pixel). A detailed discussion of the statistical analysis of the data obtained is provided in Section 4.1.3.

3.3.5 Hardware Solutions

The writing of images stored to CD-ROM will potentially change the need for any image compression requirement. It is not envisaged that this will significantly change the results obtained as there is little variability between cells in JPEG and TIFF formats (Section 4.1.2). Nevertheless, a CD-ROM writer enables the safe storing of more and better quality images, which may be necessary for any further routines developed.

3.4 AUTOMATIC COUNTING METHODS

Automatically estimating cell concentration was attempted at 200× and 1000× magnification. Absolute determination of cell concentration relies on analysing a fixed quantity of liquid and counting the cells therein. These were thus all based on the use of the Thoma counting chamber.

At 1000× magnification only one focal plane of the two which contain cells is in focus and thus the number of cells counted is not necessarily the number in the volume analysed or even half that number. This is because there is not necessarily an even distribution between the cells which adhere to the glass of the counting chamber and those which adhere to the cover slip. Nevertheless, relative results between samples should be comparable.

At 200× magnification using dark field illumination, both focal planes are approximately in focus and thus the entire volume is analysed. The size of the cells in pixels (very few pixels per cell) and the possibility of obscuring (cells on top of each other) does introduce an uncertainty in the counts. At 200× magnification the differentiation of individual cells from each other is not possible, and thus clumps of cells would be counted as one cell. This is partly overcome by estimating the number of cells in a clump from morphology data of the mean area per cell, but the 3-dimensional nature of the cells in the counting chamber ensures that this is at best a rough estimate.

3.4.1 Processing Routines Employed

3.4.1.1 Cell Counting at 1000× Magnification

These were originally attempted as a combination approach with the same images being used for the extraction of morphology and for counting. The processing routine used was

thus the same as for the morphology quantification which is discussed in section 3.5.1. The method did not prove successful and counting at 200× was employed instead.

3.4.1.2 Cell Counting at 200× Magnification

Automatic counting at 200× magnification and dark field illuminations required very little processing. This was because the cells were very bright in comparison to the background. In addition, the magnification did not allow differentiation between cellular and non-cellular bright areas. The routine therefore automatically thresholded the image on bright objects (according to the method of Otsu, 1979) with no constraint on the objective function. The number of objects (cells, clumps of cells or other bright material) and the area of each object was then extracted to an ASCII file. Errors due to extra material being reported as cells were minimal as the media was clear and the counting chamber was scratch free and thus presenting a uniformly black background. The automatic counting program code and a flowsheet of the program operation are detailed in Appendix D.

3.5 MORPHOLOGY QUANTIFICATION

The macro developed for the quantification of bacterial morphology (Appendix E) was designed to operate automatically. It therefore needed to be able to identify and quantify all potentially interesting characteristics in an image regardless of the stage in the fermentation and possible inclusions in the cells. The macro developed was followed manually through the course of the processing of the images from the bioreactor trial. It performed with a greater than 90% accuracy with respect to the complete identification of cells in the image. For a detailed discussion of the performance of the macro through various cases please consult Section 4.1.4. Appendix F details the operation of the macro for a few example images.

3.5.1 Macro Flow Diagram

The macro consisted of two nested loops (Figure 3.4, Appendix E), the outer loop stepping through each sample in turn and the inner loop stepping through each image in a sample. Processing consisted of creating templates which were then compared to the original image to locate objects (cells, spores and proteins). Figure 3.5 outlines the major processing steps within each loop of the macro.

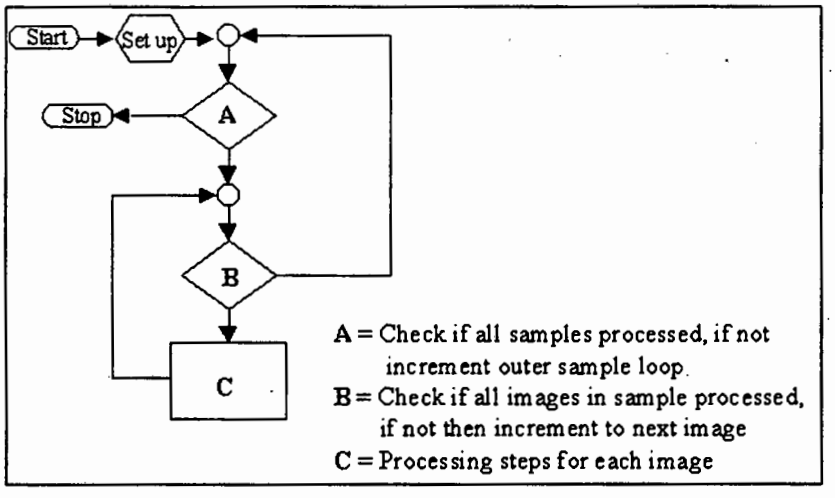


Figure 3.4: *Overall macro loop structure*

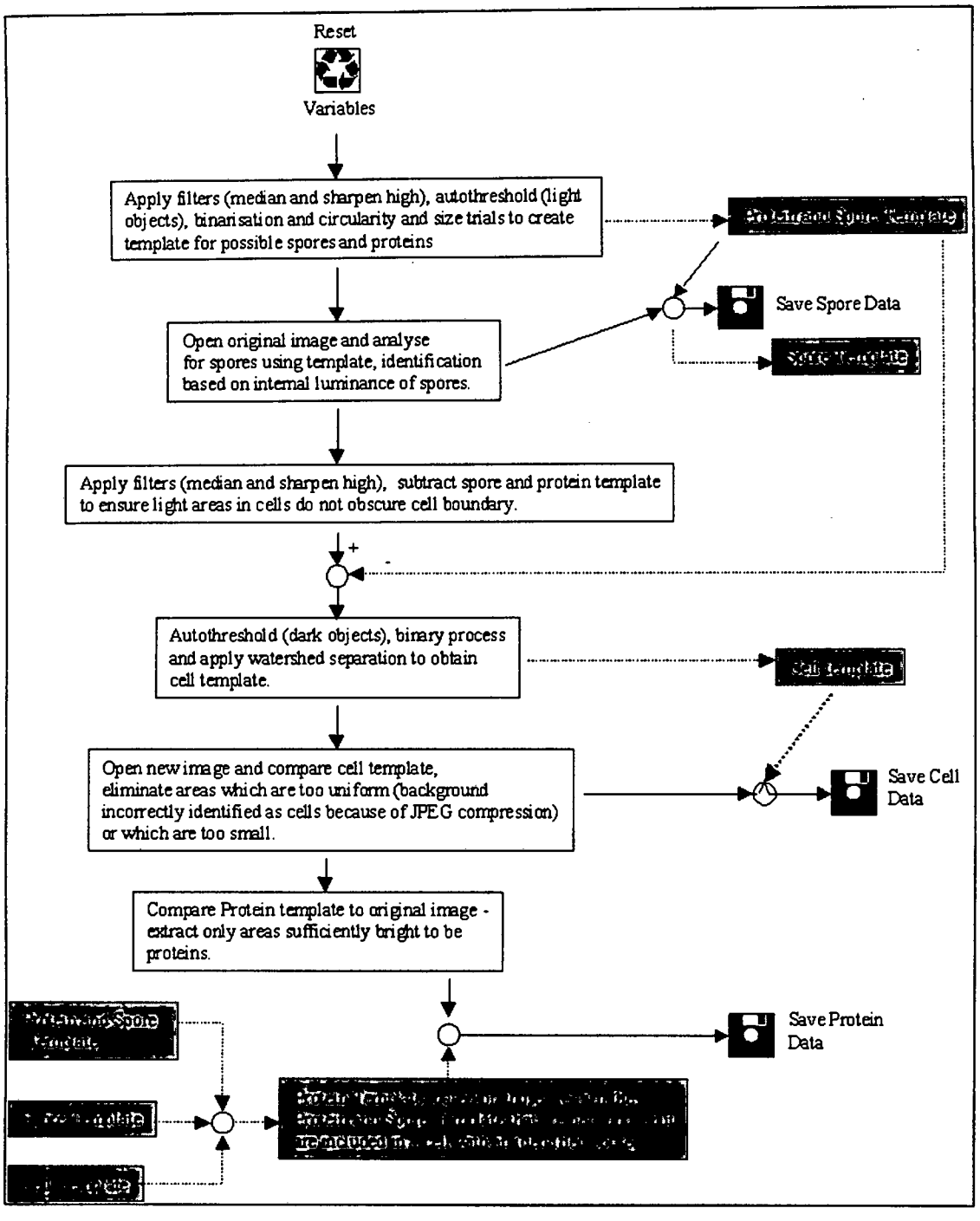


Figure 3.5: *Major processing steps within each macro loop*

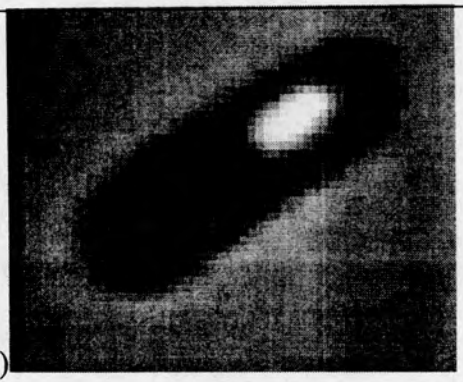
3.5.2 Filters Applied

In the stages (Figure 3.5) where the original image is processed to find light areas which could be possible spores or proteins or dark areas which could be possible cells, preprocessing is first required. This consists of applying the median and sharpen high filters to remove compression effects from JPEG compression and to highlight edges in the image.

3.5.2.1 The Median Filter

The median filter traditionally eliminates noise in an image by replacing each pixel in the original image with the median luminance value of the surrounding pixels (either in a 3×3 or 5×5 neighbourhood) in the image (Gonzalez and Woods, 1992). It is thus good at removing point-like noise in an image while still preserving edges.

A 3×3 median filter was applied in our application, but was not needed as a classical noise reduction filter as the JPEG compression scheme had already removed point like noise in the image. Furthermore, most noise is averaged out when images are acquired in colour. It was found, however, that the blocking effect caused by the JPEG compression was significantly reduced after application of the median filter. This happens as discontinuities between adjacent pixels which arise as a result of them falling into different blocks for JPEG compression are smoothed by the median filter. Figure 3.6 demonstrates the effect of applying the median filter to a JPEG compressed image.



a)

b)

a) Enlarged section of original image (JPEG). Image has been magnified 4× (16× greater area) to demonstrate the effect on individual pixels.

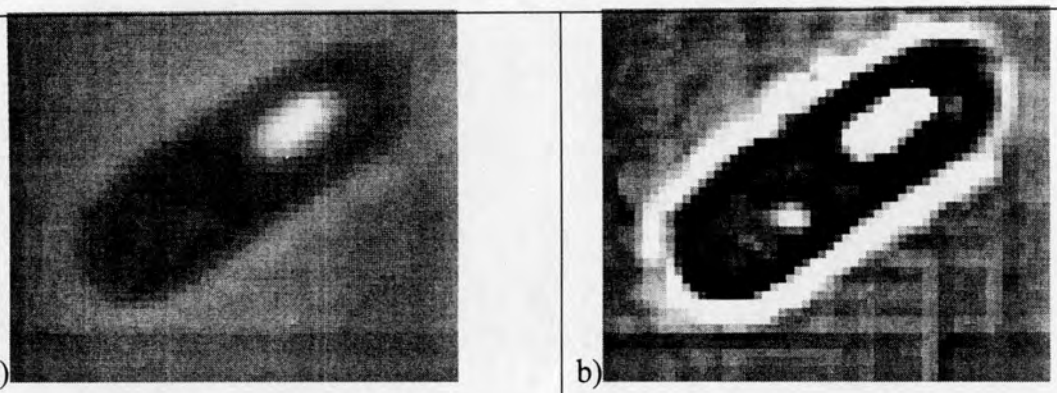
b) Same section after application of the median filter

Figure 3.6: *Application of the median filter.*

3.5.2.2 Sharpen High

The identification of regions in an image often involves detecting their edges. The "sharpen high filter" assisted in highlighting the edges in the image for later identification. Spores and crystals have particularly sharp edges as they are bright objects embedded in a dark cell.

The "sharpen high filter" is a 5×5 filter which applies a strong edge sharpening gray-scale convolution on an image. Edges tend to be highlighted in such a way that the borders of dark objects are rendered even darker and those of light objects even lighter. This was found to be beneficial for later processing as automatic thresholding and binary processing could easily identify the borders of dark or light objects. An example of the application of the "sharpen high" filter is given in Figure 3.7. Details on the use of the "sharpen high" filter are extended in Appendix G.



a) Original image after application of the median filter	b) Sharpen High filter applied. Note that the edges of the cell are now very dark, while the halo has become almost white. Notice the line at the bottom. This is a false edge introduced through the use of JPEG compression.

Figure 3.7 : *Use of Sharpen high filter*

3.5.3 Automatic Thresholding and Binarisation

Automatic thresholding on both light objects to identify possible spores and proteins and on dark objects to identify possible cells, was performed using the built in Optimas autothreshold function (Optimas, 1994b). Although not stated in the Optimas users manual, tests (Appendix H) showed that the automatic thresholding routine performs exactly as a specifically written routine based on the method proposed by Otsu (1979), with the exception that the Optimas system gives the user options to force the objective function. The objective function can be forced to consider only certain gray scale ranges or to include a minimum image area in the foreground.

In the identification of light areas which included spores or crystals (Figure 3.4), no forcing of the objective function was performed because spores and proteins are not present in all images.

Thresholding on light objects often not only highlighted spores and proteins, but also highlighted the halo seen around cells under phase contrast. These could normally be eliminated through size (large area) or circularity (high perimeter to area ratio) criteria (Section 3.5.4).

For thresholding of the image on dark objects (cell identification), the objective function of the autothreshold routine was forced to include a minimum of 1% of the image area. This was found to give a better cell boundary than when the objective function was not forced.

Once thresholded, the images were binarised with the foreground white and the background black. The binary image was then processed and analysed for objects of interest before being used to construct a template for comparison with the original image.

3.5.4 Object Classification and Identification through Shape and Area Features

In some cases the areas identified after thresholding and binarisation contain not only objects of interest (cells, spores, protein bodies) but also miscellaneous objects which have been incorrectly identified. These can then be eliminated through shape or area (Appendix C).

In the identification of possible spores or proteins for the construction of the relevant template (Section 3.5.5), all areas were first screened on the basis of their circularity and area. Any object with an area greater than 1.5 μm^2 or a circularity greater than 25 were discarded (circle has circularity of $4 \times \pi \approx 12.57$) . This was found not to eliminate any spores or proteins on manual checking, even though a circularity of 25 is quite low.

3.5.5 Templates

After binary processing and the elimination of areas based on shape or area, a template of the objects that have been located is then constructed. This can be compared to the original image and used to eliminate any areas which have still been incorrectly identified. The templates constructed and used in the macro are described below.

3.5.5.1 Spore and Protein Template

This consists of all bright areas identified through autothresholding with areas of less than 1.5 μm^2 and circularities less than 25. Also eliminated are areas which lie within 1 pixel of the image boundary (effectively a measurement frame as per Section 2.3.7). These areas are then possible spores or proteins.

The Spore and Protein template is used as a basis for finding and constructing both the individual spore and protein templates and in the identification of cells. Here the possible spores and proteins are subtracted from the pre-processed image to "black out" the bright areas within cells. This helps with thresholding and prevents bright spores in the cell from obscuring the cell boundary. It does sometimes cause the appearance of tendrils on the boundary of cells as circular blobs in the cell halo are sometimes included in the template and thus are blacked out next to the cell boundary and interpreted as being part of the cell boundary. These tendrils are largely eliminated by the binary opening described in Section 3.5.6.1.

3.5.5.2 Spore Template

The spore template is constructed from the spore and protein template through comparing the latter to the original image. Spores are identified as those areas whose internal luminance exceeds the gray-scale value of 230 (maximum internal luminance value). Overall brightness was approximately constant between images as the lamp was always

set at the present photographic setting and agar thickness was approximately constant between slides.

The spore template is used as a marker in the identification of proteins as it is used to eliminate spores when searching for proteins.

3.5.5.3 Cell Template

This is constructed from those areas which have an area of greater than 3 μm^2 and a standard deviation of greater than 3 in the histogram of their internal luminance gray scale values. The area criteria is necessary as in the later samples free spores are present and, after "blacking out" these could be incorrectly identified as cells. The use of the standard deviation in internal luminance criteria was found to be necessary as false areas are sometimes mistakenly identified as cells as a result of JPEG compression. These areas actually form part of the background and consequently have a very uniform distribution of internal gray-scale values. The cell template is used as a marker in the identification of proteins.

3.5.5.4 Protein Template

The Protein template was constructed from the Protein and Spore template, the Spore template and the Cell template. Proteins were identified only if they were areas in the Protein and Spore template which were also included in a cell in which a spore had already been identified. Stipulation of the presence of a spore in a cell before a protein can be identified is based on the sporulation dependence of the production of protein for the subspecies *kurstaki* (Agaisse and Leredus, 1995) and would need to be modified for other subspecies in which protein production is not sporulation dependent.

3.5.6 Binary Processing

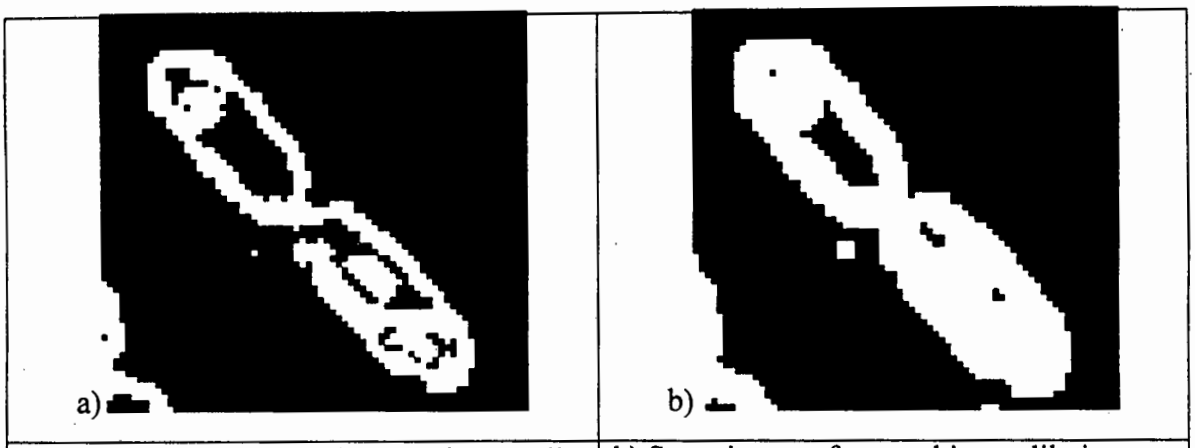
Binary processing was required on all binarised images. For possible spores and proteins, a binary fill is applied as some areas, notably proteins, had darker regions in their interiors. For cells a series of binary processes were required to ensure an unbroken cell boundary, fill identified cell interiors, eliminate tendrils and separate joined cells.

Binary erosions, dilations and binary filling operations were used for all processes except cell separation, which relied on applying the watershed separation algorithm to the inverted distance transform of the binarised image.

3.5.6.1 Binary Erosions, Dilations and Fills

All objects in the binarised image for cell identifications underwent the following processing sequence: binary dilation, binary fill, binary erosion and binary opening.

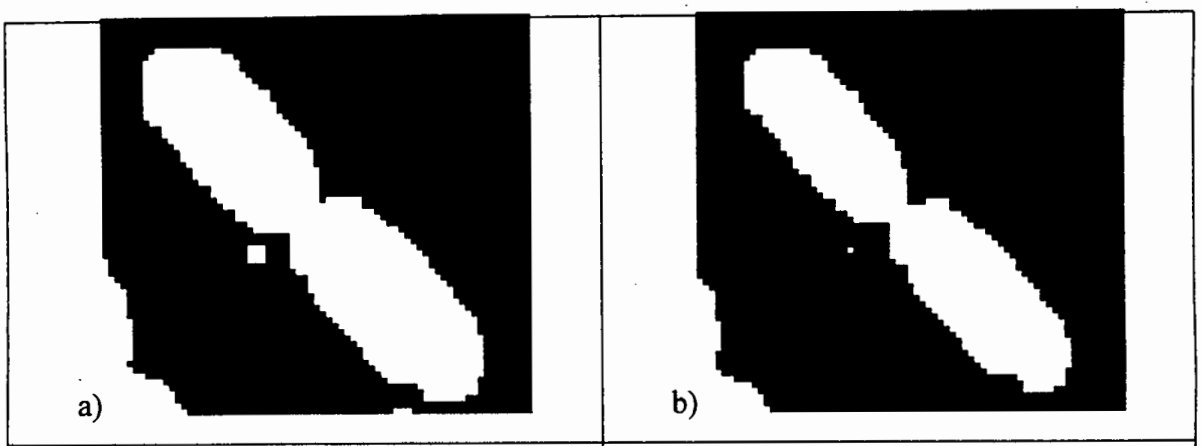
As the edges around cells were not as sharply defined as the edges around spores or proteins, a binary dilation was required. This helped restore edges which were not completely found. The binary dilation replaces background pixels in the image which are 8-connected (see glossary) to a foreground pixel as foreground pixels. This means that black pixels adjacent to a white pixel (either in the horizontal, vertical or diagonal directions) become white.



a) Enlarged (3×) binary image of two cells. Note that the edge sharpening and automatic thresholding has been inadequate in identifying the entire edge of both cells.	b) Same image after one binary dilation

Figure 3.8: *Effect of binary dilation*

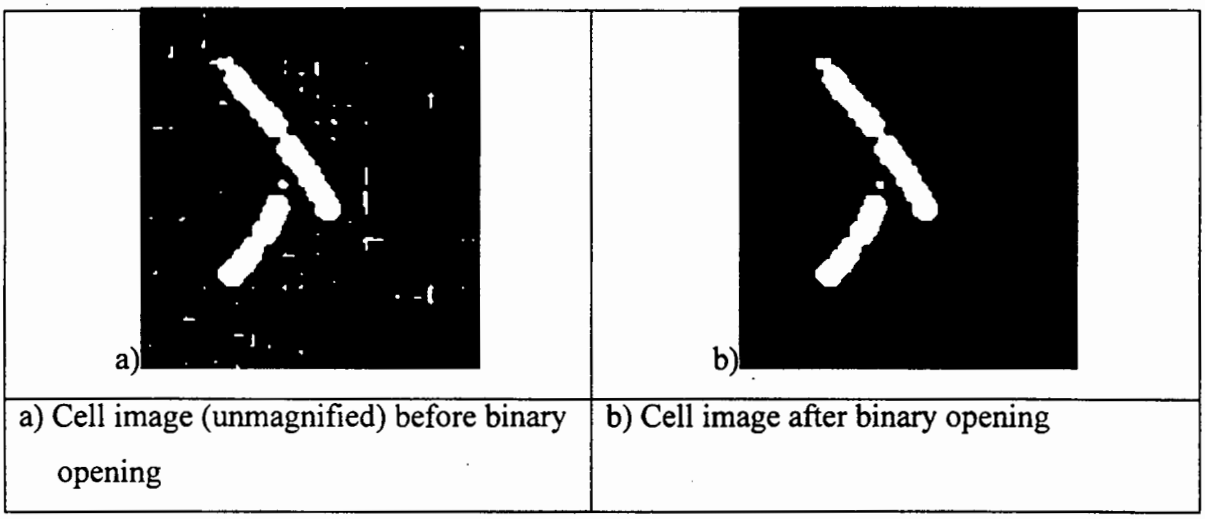
Once dilated, a binary fill and a binary erosion are performed (Figure 3.9). This restores the cell area and ensures that the entire area of the cell is now foreground (white). The binary fill simply fills in all black pixels which are completely surrounded by a white border. A binary erosion works in the opposite way to a dilation, replacing all white pixels which are 8-connected to a black pixel with a black pixel.



a) Binary fill performed	b) Binary erosion

Figure 3.9: *Effect of binary fill and binary erosion*

A binary opening was then performed. The aim of this was to remove tendrils that resulted from the earlier subtraction. A binary opening is a binary erosion followed by a binary dilation. Narrow objects get eliminated in the erosion and subsequently are not restored in the dilation. Narrow tendrils are therefore removed.



a) Cell image (unmagnified) before binary opening	b) Cell image after binary opening

Figure 3.10 : *Effect of binary opening*

3.5.6.2 Watershed Separation

Two different methods to separate joined cells were attempted. These were the use of ultimate erosion markers and the watershed separation based on the distance transform.

The method of ultimate erosion markers erodes each object in an image until that object has been reduced to a single point. This is then used to reconstruct that object separate from any joined objects which have been eroded to their own ultimate erosion marker. The method can be modified in Optimas such that the objects are only eroded a set number of times and not to their ultimate erosion point. This is presumably intended to prevent objects with irregular edges from being split.

The ultimate erosion method was not found to perform well on the binary images of cells needing separation. The number of erosions was globally set, but different cells required differing numbers of erosions to be separated. Therefore some cells would not be separated while others were over-separated. A compromise was attempted whereby only cells which were identified as possible doublet cells were processed, but this did not significantly improve the situation (Appendix I).

The application of the watershed separation method to the distance transform of the binary image proved far superior. This method is computationally expensive and therefore time consuming but is far more accurate (see Section 4.1.4 and Appendix F).

Initially the distance transform is computed for the image. This is the distance of each white pixel in the image to the nearest black pixel in the image. The distance reported is the chamfer 5-7-11 distance as this is the closest approximation to the Euclidean distance possible while still using integer gray values (Borgefors, 1986). The maximum error in approximation is only 2.02 % (Borgefors, 1986). The distance transform works by assigning each pixel with a distance value, either 5,7,11 or the smallest combination of

these values based on Figure 3.11 (e.g. a pixel horizontally adjacent to the background gets a value of 5, etc).

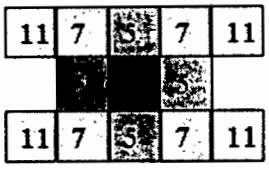


Figure 3.11: *The Chamfer 5-7-11 distances from the central pixel*

Once the distance transform has been computed the watershed separation algorithm is applied. A preflood of 9 (see Section 2.4.6) was chosen for the watershed separation. This means that a value of 9 is added to each identified minimum in the inverted distance transform. If this is higher than the watershed separating that minimum from an adjacent watershed area then that watershed area is eliminated and combined with the adjacent area. The preflood value of 9 was found to prevent over-separation as a result of irregular boundaries while adequately separating cells (Figure 3.12).

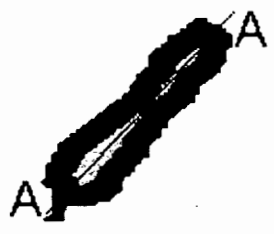
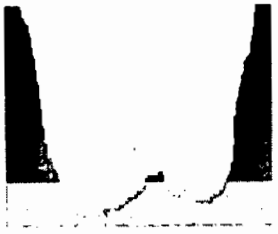
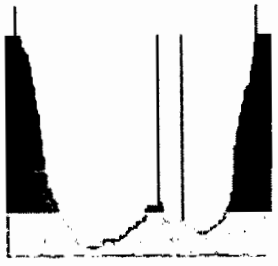
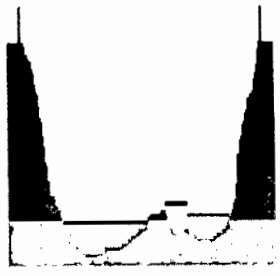
a)	b)	c)	d)
			
a) Inverted distance transform of a cell.	b) Cross sectional view through A-A	c) Watersheds (purple lines) found if no preflood applied	d) Preflood fills shallow basins, erroneous separation suppressed.

Figure 3.12: *Watershed with preflood applied to the inverted distance transform.*

The watershed separation does not, however, separate and remove tendrils or other small objects attached to the cell boundary. This is because these areas do not form deep basins on the application of the distance transform and will thus be included in the cell after preflood has been applied.

3.5.7 Data Obtained and Data Export

3.5.7.1 Data Export

All data obtained by the macro was exported to ASCII files. This was found to be quicker and safer than direct export to an open Microsoft Excel Worksheet as the ASCII files were not memory resident during the execution of the macro. They were thus preserved in the event of computer crashes, power failures or the like.

3.5.7.2 Measurements Extracted

Length, breadth, area, perimeter and average gray scale luminance were extracted for cells. An ellipse fit of the cell was also extracted. For spores, area, breadth, length and circularity were extracted. Protein area and mean gray scale luminance were extracted.

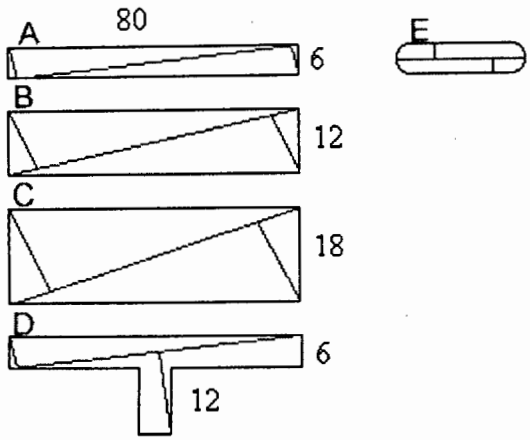
3.5.7.3 Data Processing

Data processing was performed by importing the results into an Excel worksheet and then statistically analysing the data. This included the construction of a histogram and a normal scores plot to determine the extent to which the data followed a normal distribution. Other statistical parameters extracted included the mean value, standard deviation, maximum error in the mean and the 95% confidence limits. Please refer to Section 4.1.3 for a discussion of the statistical analysis of the data.

3.5.7.4 Inaccuracies in the Data Extracted

The length, breadth and area determinations were problematic. The area extracted is based on the perimeter determinations for the area, which varied depending on the sampling method chosen. The method decided on created an area based on the boundary of an object being 8-connected and the boundary lying on the edges of the pixels. This gave the most accurate results when tested on objects of known area.

Length extracted is the major axis length, defined as the maximum distance between two points on the boundary of an object. It is not therefore the length that a human operator would measure (Figure 3.13). The breadth is the sum of the maximum distances from the major axis on either side of the major axis. It therefore can be an overestimation of the true breadth (Figure 3.13). For this reason the ellipse fit of each area was also extracted for comparison. The ellipse fit uses the centre of mass and the binary moments of the extracted area to calculate the ellipse dimensions. This tends to overestimate the cell length and breadth. See Section 4.2 for a discussion of the data extracted and variability in the data.



Examples of Optimas measurement of lengths and breadths. The major axis length of the rectangle is the diagonal length. The width is the sum of the largest distances from this (i.e. sum of the lengths of the blue lines). In the case of the bacterial shape (E) the major axis length and the "true" length should be equivalent, assuming semi-circular ends. Accuracy in the breadth determination for this shape will depend on the roughness of the area boundary.

Reported lengths and breadths: (Note that Optimas measures from centre point to centre point of the boundary pixels and therefore sees the above length as 80 – 1 pixels.)

A) Length = 79.1, Breadth = 9.9

B) Length = 79.8, Breadth = 22.0

C) Length = 80.8, Breadth = 35.0

D) Length = 79.8, Breadth = 20.0

The breadth as Optimas would report may be approximated as $breadth = 2 \cdot \frac{a \cdot b}{\sqrt{a^2 + b^2}}$

where a and b are the length and breadth of the rectangle.

Figure 3.13: *Measurement of length and breadth*

3.5.8 Macro Times to Process an Image

Total processing time was 1 minute 31 seconds per image, independent of the number of cells or other foreground objects per image. Of this, 56 seconds was for the watershed separation process (including the calculation of the inverted distance transform). Total processing time for 30 images is thus about 45 minutes. This should improve with the faster processors now available but is sufficiently fast that processing of images can be performed between sampling from a fermentation. The data generated is thus generated sufficiently quickly to give feedback about a current fermentation.

3.6 BACTERIAL MOTILITY STUDIES

The quantification of movement did not receive a comprehensive study in this work. Nevertheless, some methods were attempted and results obtained which give an indication of bacterial motility. Since it was felt that this might be an indicator of the physiological state of the organisms.

In all motility studies, the motility was determined using a 10 fold dilution of the sample in saline solution. This aided in maintaining a constant viscosity between samples. Viscosity can vary during a fermentation, possibly during the production of exotoxin and is known to affect bacterial motility (Poole, 1990).

3.6.1 Experimental Apparatus

Motility measurements were performed in the Thoma counting chamber (Weber Scientific) at 200× magnification under dark field microscopy. The measurements were taken immediately after sampling and slide preparation as motility declined with time. Qualitatively it appeared that the bacteria became attached to the glass surface. This rate of attachment may be dependent on the number of unattached bacteria.

The 200× magnification was chosen as the motile *Bt* moved too fast to observe for a sufficient length of time at 1000× magnification. The images were thus taken at a magnification where most of the chamber was in focus and where the frame dimensions were sufficiently large that the residence time of a motile bacterium within the frame was of the order of tens of seconds.

3.6.2 Image Acquisition and Data Extraction

Two different methods to quantify bacterial motility were tried. These can loosely be described as an individual and a frame based method. The individual method is based on the method described by Phillips and Quinn (1994), where a bacterium is followed manually and its motility measured. The frame based method relies on analysing the differences between images taken of the same area of the slide over a time interval.

3.6.2.1 Individual Method

This follows the motion of a bacteria over a time interval. The method is entirely manual. Initially two approaches were taken:

- Video taping the area for a time period for later analysis.
- Taking a sequence of images at 0.72 second intervals (the shortest time period to acquire and store an image) and manually determining (on the screen) which bacteria had moved and by how far between frames.

Video taping of the images is advantageous in that a moving bacterium can be followed and any tumbling, stopping or otherwise irregular motion can be monitored. It is difficult, however, to measure accurately the distance travelled as this must be done on the television screen without computer aid. This method is not reported in the Results.

The acquiring of digital images as rapidly as possible (0.72 seconds is the fastest for our arrangement) and then analysing cell motion from these has the advantage in that the distance moved in a certain time can be accurately determined. The time period between images is crucial, however, to prevent inaccuracies due to irregular bacterial motion. The ideal solution would be to acquire images at 20 – 25 frames a seconds (the rate that a video camera records) for later digital analysis.

Only the digital image acquisition method was tried and results are reported but no attempt was made to extract sufficient data to correlate bacterial motility to any external parameter.

3.6.2.2 Frame Based Method

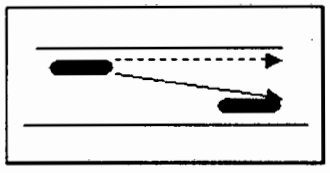
This method relies on the assumption that an image will remain the same over a period of time with the exception of any motile bacteria, which will have moved. Thus the comparison of two images should give an indication of the number of motile bacteria as a proportion of the total. This is assuming that the images are taken at a sufficient time period apart to ensure that all motile bacteria would have moved by at least 1 bacterial length. The method does not make any assumptions about the distribution of bacterial speeds but merely which bacteria are motile and which are not.

Ten images are taken at 0.72 second intervals over 7 seconds. These were then used both for the individual method and the frame based method. For the frame based method the image was binarised after automatic thresholding and a binary Xor between succeeding images was applied (i.e. those areas which are white in both images become black, those which are black in both images remain black and those which are white in one image and black in the other become white). The resultant image (binary Xor'ed image) should then have twice the white area of the motile cells in the original image. A comparison of the white areas in the original and resultant image then yields the percentage of cells which are motile (based on area).

3.6.3 Shortcomings in the Methods

3.6.3.1 Individual Method

Problems with the method where each cell is followed over 10 frames in 7 seconds stem from the fact that the apparent path length of the bacterium between frames is dependent on the bacterium maintaining a horizontal and straight swimming pattern (Figure 3.14).



(Dotted line the apparent distance, solid line real distance)

Figure 3.14: *Underestimation of real motility*

The method will therefore always underestimate the real bacterial motility. It is also highly labour intensive and requires many hours of work to yield statistically significant data.

3.6.3.2 Frame Based Method

This method is based on assuming that the number of bacteria which move into the frame in the time period between image acquisition (0.72 seconds) would equal the number leaving the frame and that this is a small percentage of the total. It also assumes that there is no bulk fluid flow in the counting chamber and that the influence of Brownian motion is negligible. Bulk fluid flow is generally not observed if the slide preparation is properly performed. If bulk fluid motion was observed, a new slide was prepared before measurements were taken. Brownian motion was not apparent for *Bt*, probably because the cells are rather large. It would definitely be a problem for smaller bacterial species but should still not influence attached bacteria.

3.7 THE RADON TRANSFORM

The Optimas version 5.2 software does not contain any shape based object identification systems and thus cannot automatically process the Radon or Hough transforms. It was felt nevertheless that the application of such a transform may be the best method to classify cells. Shape information is inherently found using this technique. This would, for instance, improve the estimation of bacterial lengths and breadths.

An Optimas macro and a Visual Basic program were developed to apply the Radon transform to partially processed images. In addition the visual basic program included an application of the Hough transform for circles (the Hough transform for lines and the Radon transform with one iteration are equivalent).

Please consult Appendix J for a detailed coding of the macro and the Visual Basic program and the attached CD-ROM for an executable version of the Visual Basic program.

3.7.1 Preprocessing Required

All processing required to completely identify the cells was first performed on the image. These procedures are the same as for the standard Optimas macro. After the cells were completely identified the image was binarised and then a binary outline operation was applied. This leaves a 1 pixel wide outline of the cell in white, the rest of the cell is black.

3.7.2 Major Steps in Optimas Macro and Visual Basic Program

The coding of the Radon transform required the construction of transform space matrices which could be used to find the probabilities of straight lines occurring in the original image. These matrices could then be searched for parallel straight lines of similar length and of a distance apart which was similar to a bacterial breadth. This would indicate the presence of a bacterium and give the breadth and length parameters for the bacterium, using the assumption that the bacterial shape could be approximated by parallel lines with semi-circular ends.

The Visual basic program (Figure 3.15) first constructs an accumulator matrix. This is done by locating each white (edge) point in the original image and accumulating (adding one to) a series of points in the transform space based on:

$$p_j = x_i \cos\theta_j + y_i \sin\theta_j \qquad (3.1)$$

Where the i and j subscripts refer to ordered pairs in the image and transform spaces respectively. For every point (x_i, y_i) of the image, i is fixed and the values p_j are calculated using stepwise increments of θ_j (Leavers, 1992b). The transform space thus generated is 2-dimensional, with dimensions in θ and p (i.e. polar co-ordinates). It's size is dependent on the size of the increments of θ. In the program developed θ is incremented in 1° intervals over the 180° required. This gives a transform space of size 180 by the maximum distance of any pixel in the original image from the origin, where the origin is defined as being in the middle of the image.

Thus after one white, or edge, point has been identified in the original image and accumulated into the accumulator matrix, some points in the accumulator matrix will have a value of 1, whereas the rest will have a value of 0. After two white points have been accumulated, there will be some points in the accumulator matrix with a value of 2, some with a value of 1, and the rest with a value of 0; and so on. Once all white points in the original image have been processed, the points with the highest values in the

accumulator matrix will correspond to the highest probabilities of lines (i.e. the θ and p values of lines) in the original image.

Both the Optimas macro and the Visual Basic program construct an accumulator matrix and then search this matrix for regional maxima. The maxima found are then compared with each other to ensure that they do not correspond to the same line (determined by checking that lines with a similar θ value do not have p – values within 3 pixels of each other). Once unique lines have been found these are sorted to determine which lines are parallel and of a distance apart similar to a bacterial breadth (defined as similar θ values of within 3°, p – values of greater than 10 pixels apart but less than 30 pixels apart and the lines must be of a similar length). These are then classed together and a bacterium is thus "found". The bacterium found would thus have a length equal to the average length of it's two edge lines plus the distance between the two lines, and a breadth of the perpendicular distance between the two lines. These measurements are easily found as the equation describing the line is found by the program (the θ and p values) and the line length corresponds to the value of the maxima in the accumulator matrix (Figure 3.16).

3.7.3 Comparison of Optimas Macro and Visual Basic Program

Both the Optimas Macro and the Visual Basic program are extremely time consuming as they analyse each pixel in the original image and in the accumulator matrix when processing. Execution of the Visual Basic program is of the order of 10's of minutes per image (size 760×512 pixels), while the Optimas macro is considerably slower than this. Methods to decrease the processing times of the two programs were not attempted as this was not the main aim of the project. It is felt, however, that with better programming algorithms, the speed of execution should be considerably reduced.

3.7.4 Application of the Hough Transform for Circles

In addition to searching for and finding lines in an image, the Visual basic program was extended to construct a 3-dimensional transform space to find circles in an image. These should correspond to the ends of bacteria (assuming semi-circular ends) and their diameters should be similar to the breadths of the bacteria. The transform space constructed has dimensions in x,y (the circle centre points) and r (the circle radius). The transform space constructed here was given a maximum size in the r dimension of 20 pixels (i.e. no circles with diameters greater than 20 pixels would be found) to decrease it's size and thus it's memory requirements and the time to process.

The circle procedure than finds circles in the image (maxima in the transform space) and determines their uniqueness (centres and radii sufficiently different). Once this has been performed the circles are checked with the lines found in the earlier transform for lines to find the bacterial "end circles". This should give extra evidence of a found bacterium as the circle diameter should correspond to the bacterial width and the distance between the circle centres plus a circle diameter should correspond to the length of the bacterium.

The routine is around 20× slower (the value of the r dimension) than the routine for lines. It was thus only used on small sample images (see the attached program on CD). Again, more advanced algorithms should improve this time, but it is felt that only advanced and expensive computer systems will make the use of such a routine viable.

4. RESULTS AND DISCUSSION

Since the primary aim of this thesis is to develop a robust algorithm for the image analysis of *B. thuringiensis* it was decided that the validation of the image processing methods chosen should receive the most attention.

Results are reported and discussed in 3 separate sections. Section 4.1 covers the rigorous validation of the methods and sequences chosen. This includes procedures such as slide preparation, the number of images acquired and the image processing steps employed. Section 4.2 illustrates the application of these methods and sequences to the morphological evaluation of *B. thuringiensis*, the type of data obtained and interesting trends observed. This serves to give an indication of possible outputs expected from a comprehensive study employing the methods developed in the project, but is not comprehensive itself. Section 4.3 discusses preliminary work on bacterial motility. This work gives an indication of what approaches can be taken when quantifying bacterial motility.

4.1 IMAGE ANALYSIS METHOD VALIDATION

Validation of the methods used was directed at verifying the image acquisition, the storage phases and the image processing.

4.1.1 Use of Water Agar

Owing to difficulties discussed (Section 3.2.3) with respect to maintaining cells in the focal plane on observation of a wet preparation, the use of water agar slides was tested. The immobilisation of cells on water agar as a means of taking images for morphology measurements was aimed at ensuring that all bacteria and cell inclusions (spores and protein crystals) remained in focus and in the same orientation (i.e. are lying flat) relative to the objective lens of the microscope. To ensure that this slide preparation did not result in morphological aberrations, images were also acquired using a Thoma counting chamber (Weber Scientific) and these were compared to the water agar images. Table 4.1 shows the reported mean length and breadth measurements (for 90 bacteria) extracted from a fermentation sample using images acquired from cells in both the Thoma counting chamber and those mounted on agar.

Table 4.1: *Differences between cells measured on agar and in a counting chamber*

Property	Agar	Counting Chamber
Mean length at 4 hours	7.25 µm	6.78 µm
95% confidence level in mean length at 4 hours	0.54 µm	0.45 µm
Mean breadth at 4 hours	1.80 µm	1.73 µm
95% confidence level in mean breadth at 4 hours	0.06 µm	0.09 µm
Mean length at 24 hours	4.67 µm	4.66 µm
95% confidence level in mean length at 24 hours	1.32 µm	1.38 µm
Mean breadth at 24 hours	1.90 µm	1.69 µm
95% confidence level in mean breadth at 24 hours	0.07 µm	0.05 µm

Images acquired on water agar should show less variation in reported bacterial length than images acquired using the Thoma counting chamber as cells in the latter case can be in a number of different orientations and focal planes. This was not found to be the case (Table 4.1). Table 4.1 shows that images acquired of cells on water agar and of cells in the counting chamber showed no significant differences in reported mean cell length or breadth. A manual check on the image processing steps showed that these were sufficiently robust to eliminate cells which were out of focus in images taken of cells in the counting chamber. Only cells adhering to the glass of the counting chamber were therefore counted. These cells were both in focus and orientated correctly with respect to the camera.

It is useful to stress that the location of other types of bacteria and the quality of images produced is markedly improved when using agar to mount the bacteria. In particular, observation and quantification of bacterial clump sizes of *Corynebacterium glutanicum* have been solved through mounting the bacteria on water agar (C. Henwood, personal communication).

For the correct location of intracellular spores and crystals from images acquired on agar and in the counting chamber, there is a marked difference in the results obtained. A sample of well sporulated cells was analysed and counted both automatically and manually in the counting chamber and on agar. The results are reported in Table 4.2.

Table 4.2: *Differences in the location of spores and protein crystals for agar and counting chamber images (manual validation)*

	Total spores manually located	Total spores automatically located	Spores missed	Areas incorrectly identified as spores
Agar mounted	53	51	4	2
Counting chamber	51	36	17	2
	Total Protein crystals manually located	Total Protein crystals automatically located	Protein crystals missed	Areas incorrectly identified as Protein crystals
Agar mounted	50	40	10	0
Counting chamber	45	21	24	0

The results reported in Table 4.2 indicate that the use of agar is essential if meaningful data on spores and crystals is to be extracted. The location of a protein crystal within a cell is dependant on the presence of a spore in that cell. This sets a maximum limit to the accuracy of extraction of protein crystal data. This limit is determined by the accuracy at which spores can be located in an image. Thus, the decreased accuracy in finding spores in a counting chamber image has a direct impact on the number of protein crystals found.

4.1.2 JPEG Storage of Images

The storage of images compressed under the JPEG image compression method is not ideal as JPEG compression is a lossy compression method. Nevertheless, earlier samples were obtained before the acquiring of a CD-writer and hence JPEG compression was necessary due to system memory constraints. Images were later written to CD and stored as 8-bit RGB TIFF images (1.3 Mb each). They thus did not lose any information on storage.

In order to check on the extent to which JPEG image compression introduces variability and inaccuracies in any data obtained, a full shake-flask trial (30 images per sample, 7 samples in the trial) was processed where the same set of images was stored in both TIFF and JPEG formats. The results obtained were then analysed for variability in the reported length, breadth and area measurements and in the total number of bacteria and spores correctly identified.

4.1.2.1 Total Cells Identified

In Figure 4.1, the number of cells located in each image set stored as TIFF and JPEG files is compared across a growth curve of *B. thuringiensis*. Only a slight variation (less than 10% if the total number of cells found) in the total number of cells found per image was recorded. Generally, cells which had poor edges were not identified in the JPEG images. These poor edges were due to the cell being slightly out of focus. In a few cases, large areas on the agar were incorrectly identified as cells in the JPEG images. This occurred through the introduction of false edges as a result of the JPEG compression method (Section 3.3.3).

For all the samples (7 in the whole trial), 954 bacteria were identified in the TIFF images and 920 in the JPEG images. Of the 920 identified bacteria in the JPEG images, 914 of these corresponded to bacteria identified in the TIFF images while 6 were false areas introduced through blocking. Hence there was a 95.8 % correlation between bacteria identified in the JPEG images and those identified in the TIFF images.

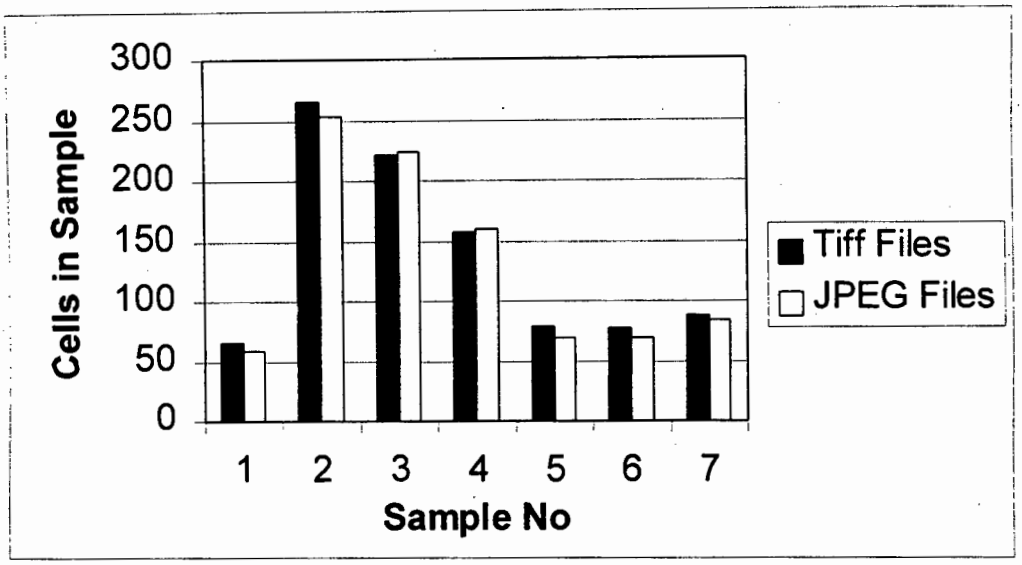


Figure 4.1: *Comparison of Number of cells located in TIFF and JPEG images*

4.1.2.2 Comparison of Cell Dimensions

A comparison of the reported mean and standard error of the area, length and breadth dimensions for the images as processed in TIFF and JPEG format showed very little difference between the two storage formats (Figures 4.2, 4.3 and 4.4 respectively). The only significant change observed was in the reported length at 9 and 13 hours. These samples each contained 3 large "false" objects which affected the reported mean length. These "false" objects actually form part of the background but have been identified as cells as the JPEG compression introduced false edges in an otherwise uniform background. This is shown by the standard error (vertical bar) and can be eliminated by removing the top 5% of reported lengths (See Section 4.1.3).

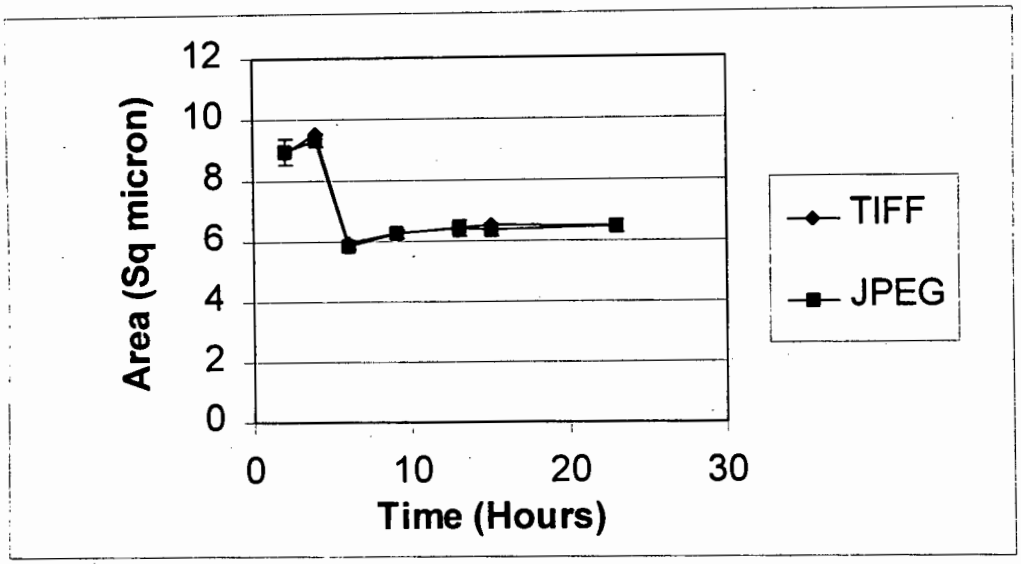


Figure 4.2: *Comparison of mean areas determined based on TIFF and JPEG storage of images*

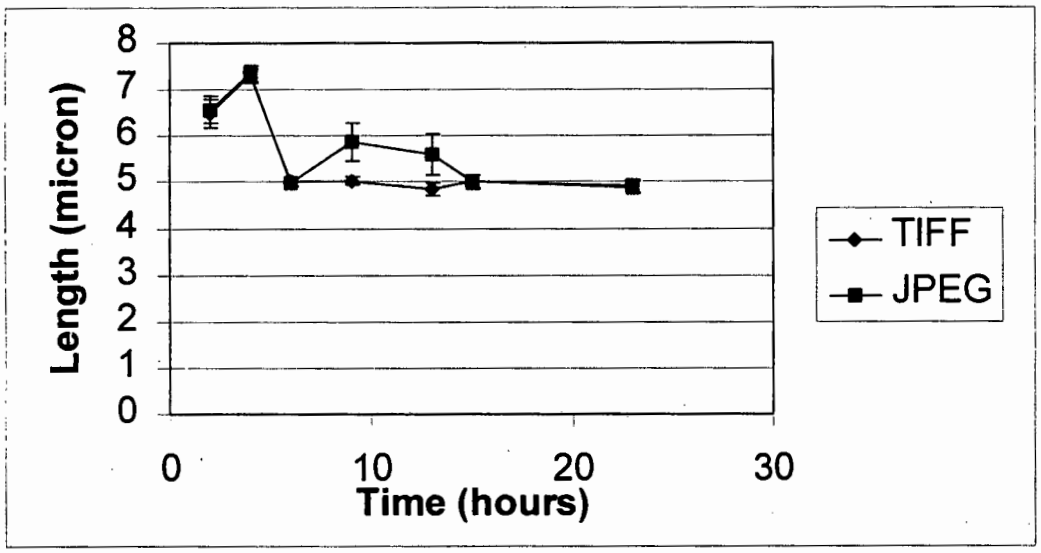


Figure 4.3: *Comparison of mean lengths determined based on TIFF and JPEG storage of images*

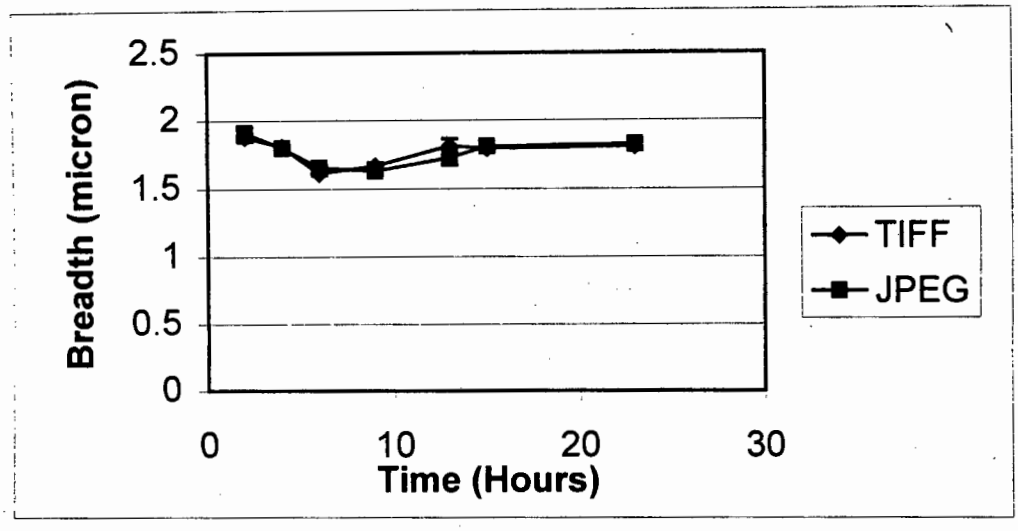


Figure 4.4: *Comparison of mean breadths determined based on TIFF and JPEG storage of images.*

4.1.2.3 Correct Identification of Spores and Proteins

The number of spores identified in the JPEG and TIFF formats was almost identical (Table 4.3). Only images at 2, 17 and 21 hours were compared since *Bt* only produces spores in the stationary phase (spores in the 2 hour image are those still present from the inoculum).

Table 4.3: *Identification of spores in images stored as TIFF or JPEG files.*

Hour	Number of Spores found		Mean Spore Area		Standard error in Area	
	JPEG	TIFF	JPEG	TIFF	JPEG	TIFF
2	26	25	0.93 µm	0.90 µm	0.07 µm	0.06 µm
17	13	13	0.73 µm	0.72 µm	0.09 µm	0.10
21	74	77	0.75 µm	0.74 µm	0.03 µm	0.03 µm

4.1.3 Number of Images Acquired per Sample

The number of images acquired per sample is a balance between :

- The time required to acquire and process the images.
- The memory requirements of storing the images.
- The reproducibility and accuracy of the data obtained.

Any practical application of the developed method would require a compromise between these requirements. The higher the number of images acquired, the greater the accuracy in the mean values reported since for large samples the sample standard deviation approaches the population standard deviation (Miller *et al.,* 1990). The standard error of the mean is related to the number of bacteria analysed (which in turn is related to the number of images acquired) by an inverse square relationship. Doubling the sample size will only decrease the error in the mean by a factor of $\sqrt{2}$. Above a certain sample size, the large increase in processing time for a relatively small decrease in error does not warrant acquiring and processing more images (Table 4.4).

The size of the error is related to the degree to which the sample distribution in the mean approximates a normal distribution. This can be seen more easily by constructing a normal scores plot of the sample data. If this is linear, the population is approximately normally distributed and the error in the mean can be calculated using the sampling distribution in the mean (i.e. the population standard deviation is unknown but is approximated by the sample standard deviation; Miller *et al.,* 1990).

Normal scores and histograms for cell length from a number of different images of the same sample were analysed and are reported below. Ninety images of a sample were taken in total and a statistical analysis was performed on the results obtained when analysing progressively more images. Figures 4.5 to 4.10 show the normal scores and histograms for the analysis of sample sizes of 10, 30 and 90 images.

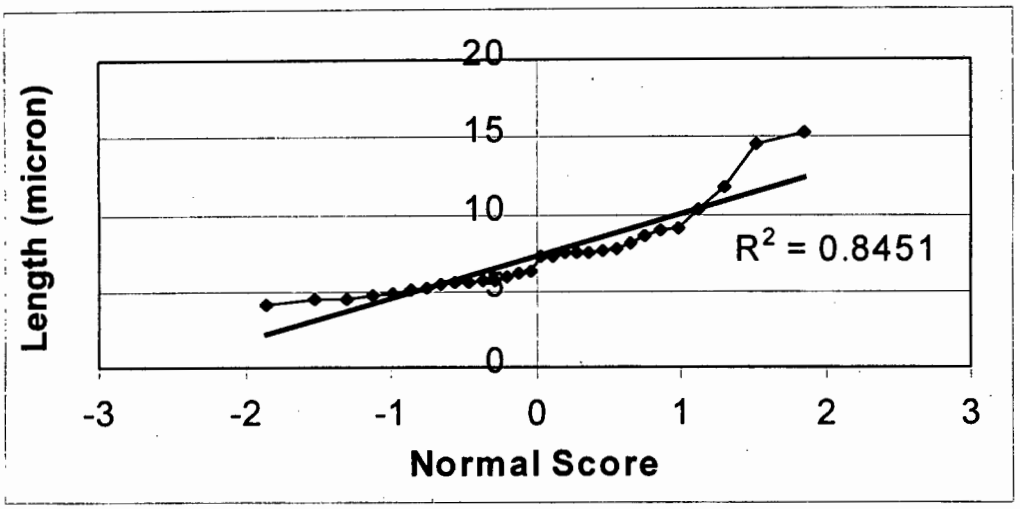


Figure 4.5: *Normal Scores Plot for 10 images (31 cells)*

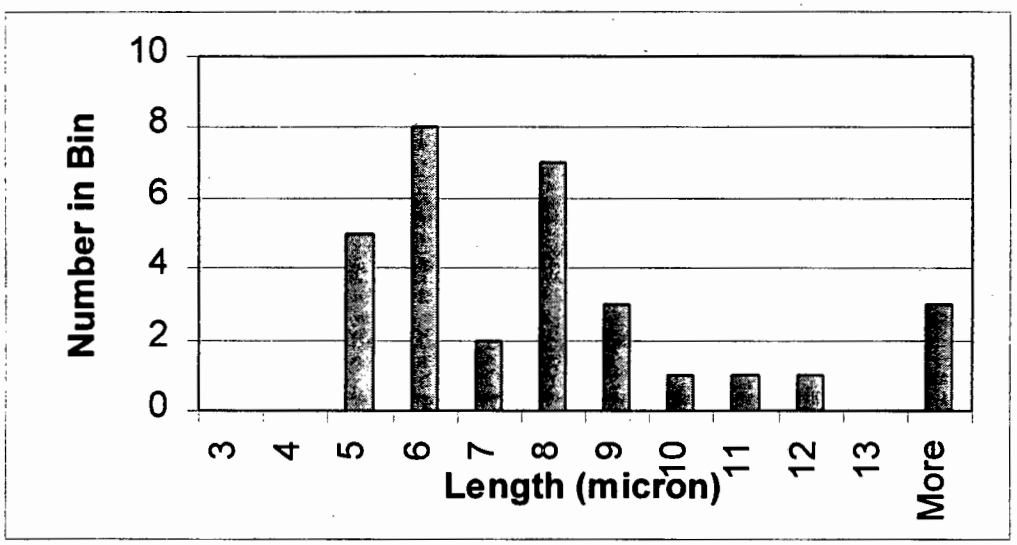


Figure 4.6: *Histogram of Length Distribution (10 images, 31 cells)*

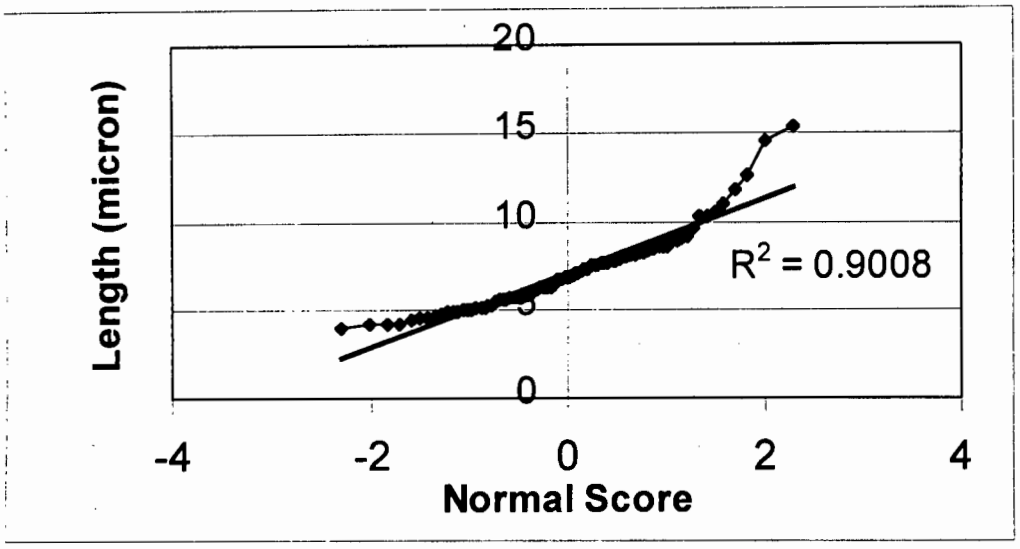


Figure 4.7: *Normal Scores plot for 30 images (90 cells)*

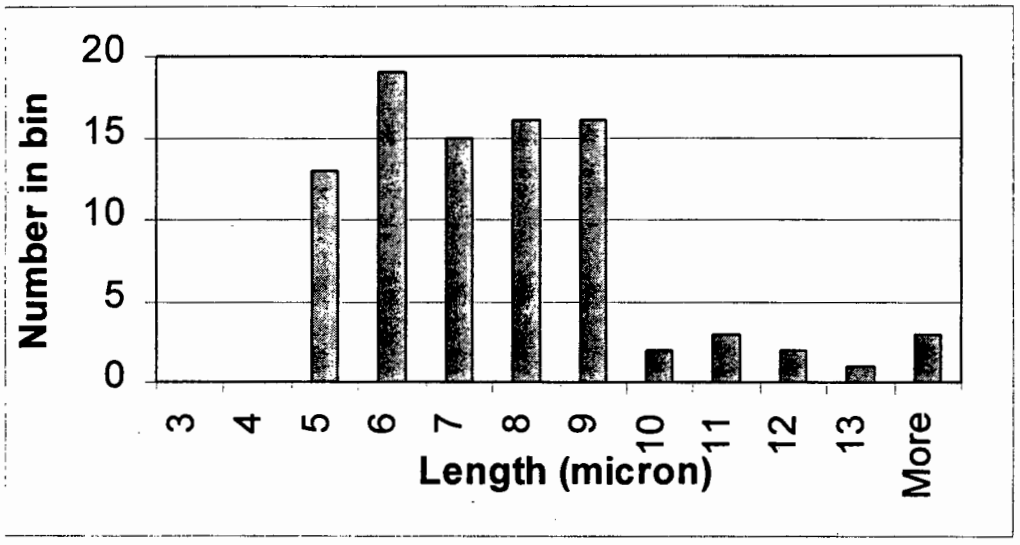


Figure 4.8: *Histogram of lengths for 30 images (90 cells)*

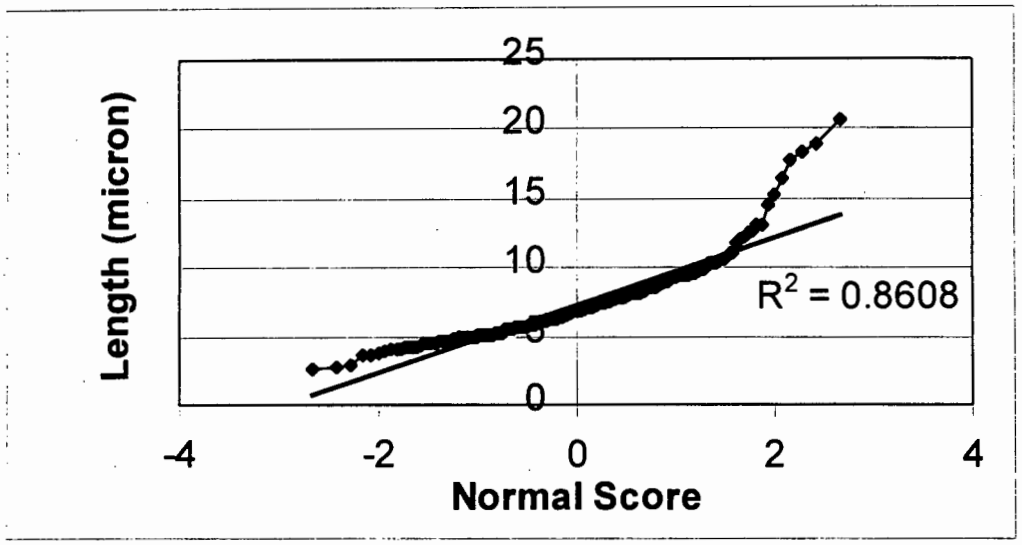


Figure 4.9: *Normal Scores Plot for 90 images (266 cells)*

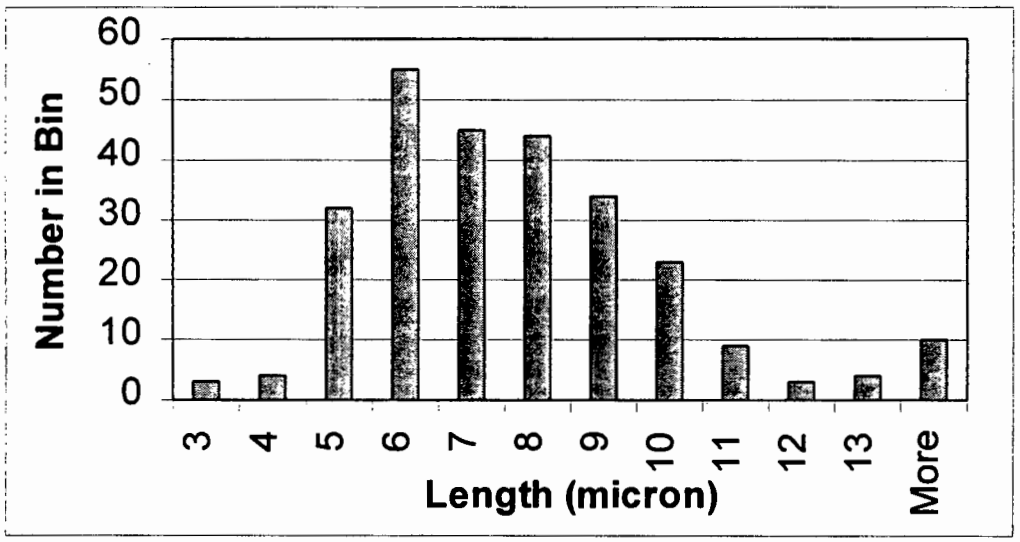


Figure 4.10: *Histogram of Lengths for 90 images (266 cells)*

The normal score plots (Figures 4.5, 4.7 and 4.9) show the closeness of fit in the middle regions of the graph, with two "tails" at the ends. The lower "tail" (corresponding to shorter cells) is expected since a normal distribution implies that there is no lower limit to cell length. As cells have some minimum length, the smallest cells are too large for the normal distribution (i.e. they sit above the fitted line). The larger objects, which are too large for their corresponding position on a normal distribution, result either from incorrectly identified areas or from cells which have not been separated. These correspond to approximately 5% of the objects found by number. If the top 5% of objects found are removed from the sample, the normal scores plot then shows a far better linearity (Figure 4.11).

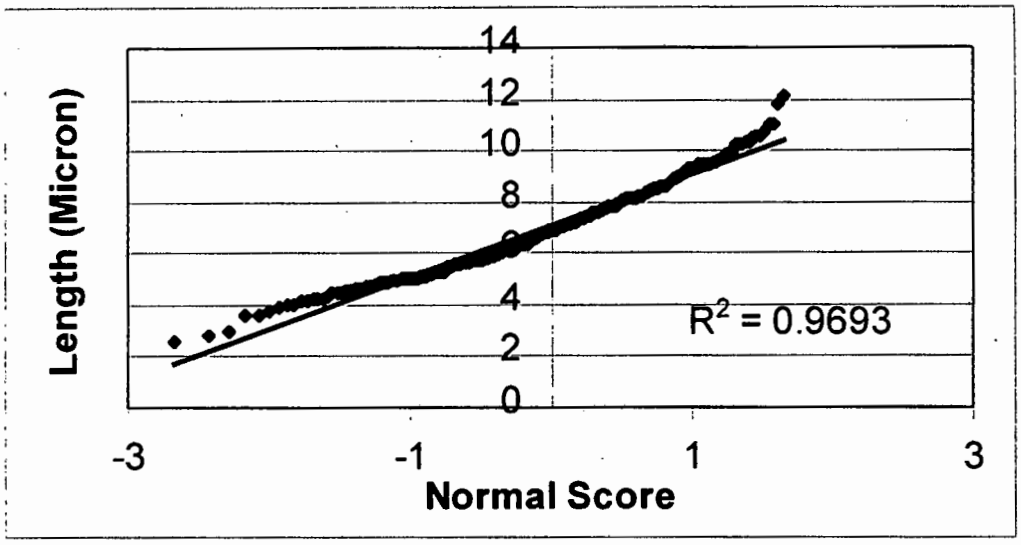


Figure 4.11: *Normal Scores Plot for bottom 95% of sample of 266 cells*

Comparison of the 95% confidence interval in the mean for the different number of cells analysed, showed a narrowing of the 95% confidence interval in the mean cell length from 1.33 μm for 31 cells to 0.33 μm for 266 cells (90 images). The confidence limits for 90 cells (30 images) was 0.54 μm (Table 4.4). The change from 30 images to 90 images of only 0.21 μm was considered insufficient to justify acquiring more images. Each image takes 1 minute 45 seconds to process plus additional time to acquire and store. The extra time required to process more images (about 2 hours extra to process 90 images instead of 30) will impact on the rate at which data is extracted and therefore on the rate of feedback to an industrial process.

Table 4.4: 95% Confidence intervals for an increasing number of images acquired

Number of images acquired	Number of cells processed	Mean Length reported	95% Confidence limit in mean length reported
10	31	7.72 µm	1.33 µm
20	64	7.28 µm	0.70 µm
30	90	7.25 µm	0.54 µm
60	191	7.16 µm	0.35 µm
90	266	7.30 µm	0.33 µm

4.1.4 Manual Validation of Results Obtained

Validation of the image analysis methods employed was performed by comparing the systems object recognition to an operator's manual interpretation and classification of objects in an image. This was carried out for the bioreactor trial. All images were taken as wet mounts on water agar slides. As a comparison samples were taken from a shake-flask trial in which a greater number of protein counts had been obtained (Table 4.6). This may have been because nutrient broth had been used rather than fermentation media. The results of this exercise are reported in Tables 4.5. Please also refer to Appendix M for a detailed comparison of each image analysed both manually and automatically.

The "correct" number of cells found manually included cells in which a tendril had been added to the cell border in image processing (Section 3.5.5.1). The presence of a tendril will not significantly affect the reported area and is unlikely to affect the reported length as the tendrils are most often located on one of the sides of the bacterium.

Table 4.5: *Manual validation of automatic cell and spore identification (Bioreactor trial, fermentation media)*

	Hours						
	5	7	9	13	16	18	21
Total Cells (Manual)	131	375	201	182	105	99	
Cells correct (including tendrils)	122	339	190	174	100	91	
Cells not found	9	36	11	8	5	8	
Areas not cells but identified as such	2	5	6	4	5	5	
Tendrils influencing breadth	8	21	14	5	9	6	
Total spores (manual)				14	87	68	35
Spores correct				13	85	63	35
Spores not found				1	2	5	0
Areas incorrectly identified as Spores				1	0	0	0
% Cells found correctly	93.12	90.4	94.5	95.6	95.2	91.9	
% Spores found correctly				92.9	97.7	92.6	100

Table 4.6: *Manual validation of automatic spore and protein identification for shake-flask, final sporulated sample at 24 hours. (Nutrient broth, 1 litre shake flask, see Section 3.1.5)*

	Total manually identified	Total automatically identified	Number not found	Number incorrectly identified
Spores	53	51	4	2
Protein crystals	50	40	10	0

In all cases at least 90% of in an image were identified. Errors in ascribing extraneous objects as cells, spores or proteins did not exceed 5 % of the total number of objects found. These figures compare favourably to those reported by O'Shea and

Walsh (1996), whose semi-automatic yeast classification system had errors of 5.6%. The system described in this thesis is faster than that of O'Shea and Walsh and is completely automatic. In addition, it is required to completely identify objects and objects within objects (spores and protein crystals) and not merely classify them into different shape classes. Protein identification performed at 80% as it was dependent on correct spore identification prior to protein identification.

4.1.5 Automatic Counting Procedure

The automatic counting procedure was compared to manual counts over different areas of the counting chamber, and to dry mass determinations. While it was recognised that using the same counting chamber preparation requires automatically counting with a thin coverslip which introduces added error (Section 4.1.6), it was needed in order to directly compare counts performed automatically and manually.

Automatic counts correlated well with manual counts (Figures 4.12 and 4.13). The automatic counting procedure used auto thresholding on bright areas and hence correctly identified possible cells (bright as viewed under dark field illumination). As the magnification was low (200×), the cell areas found were small and would not be found under the default area settings of Optimas version 5.2. This default setting requires a minimum area size such that the perimeter of the area is at least 10 pixels. This was decreased to 3 pixels for these counts. The total numbers of areas identified were then assumed to be cells and were correlated directly to the manual counts.

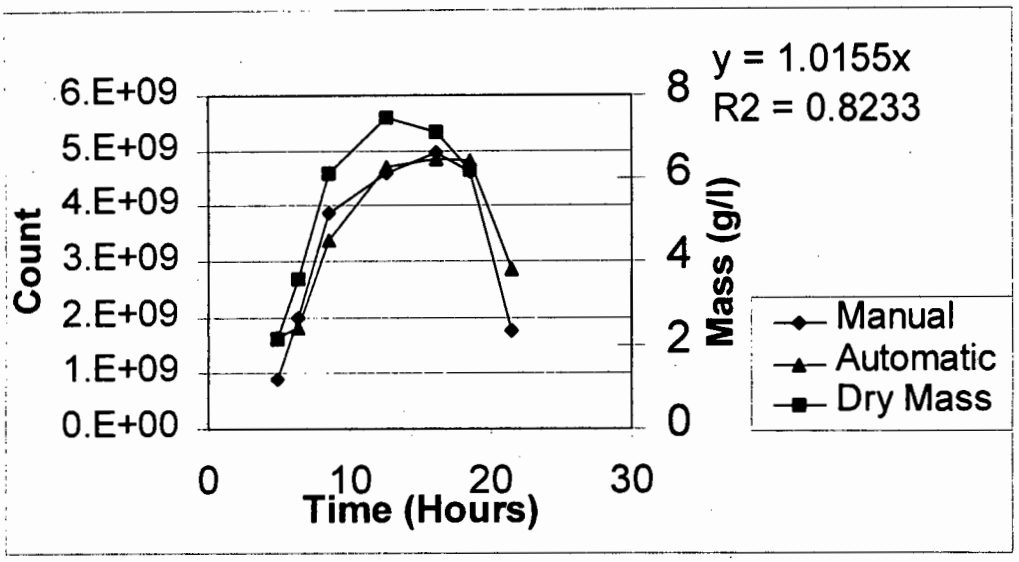


Figure 4.12: *Comparison of automatic counting, manual counting and dry mass for the bioreactor trial.*

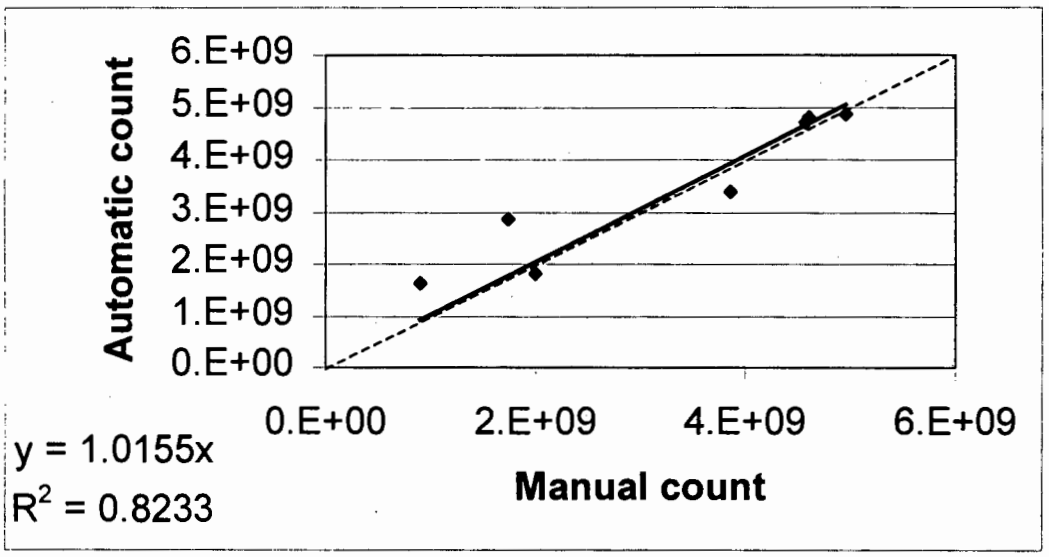


Figure 4.13: *Correlation between manual and automatic counts for bioreactor trial (Dotted line 1:1 correlation)*

It is important to note that the area finding technique cannot differentiate between a single cell and two or more joined cells due to the low magnification used. Thus many doublet cells will be incorrectly identified as single cells. It was anticipated that the automatic counting procedure should therefore underestimate the total numbers. This was not found to be the case, particularly since samples were diluted and then

agitated, which tended to prevent cell agglomeration. The deviation from parity (comparison between dotted and straight line in Figure 4.13) is not known, although it could be related to the position of the area counted in relation to the edges of the counting chamber. The manual counts were performed above the grid of the counting chamber and thus directly in the middle of the counting chamber. The automatic counts were necessarily performed above the areas of the counting chamber that were not covered by gridlines, and thus were closer to the edges of the counting chamber. As thin coverslips distort (Section 4.1.6) it would be expected that the height in the middle of the coverslip would have the greatest variation from the height at the edge. The volume calculated using the counting chamber height will thus be larger than the true volume (as the coverslip will be closer to the gridlines). It follows that the manual count should report lower bacterial numbers as the volume over which counting has been performed is lower than that for the automatic counts.

The area covered by the 5 counting images is 3.84 times larger than the area manually counted. This factor is introduced for area correction.

Biomass correlation to Automatic and manual counts is reasonably good. The decrease in biomass concentration towards the end before the drop in bacterial numbers is due to a decreasing bacterial area and hence volume (Appendix N, Figure N.4).

4.1.6 Use of a Thin Coverslip vs a Thick Coverslip

The use of a thin coverslip in manually counting cells at 1000× magnification is necessary given the depth of focus of the objective lens at this magnification. A thin coverslip can, however, distort in slide preparation and can thus alter the volume of the counting chamber. 1000× magnification is required for manual counts to discern the difference between cells and it is thus necessary to quantify the effect of a thin coverslip on any counts obtained.

Automatic counts of slide preparations from the same sample, prepared on the Thoma counting chamber (Weber scientific) using thin and thick coverslips, were compared. These and the manual counts from the same sample are detailed in Table 4.6. Table 4.7 details the mean and standard deviation for this data.

Table 4.7: *Comparison of automatic and manual counts using thick and thin coverslips*

Slide number	Automatic count on thick Coverslip	Automatic count on thin coverslip	Manual count* on thin coverslip	Slide number	Automatic count on thick coverslip	Automatic count on thin coverslip	Manual count* on thin coverslip
1	-	-	607	7	617	-	-
2	-	315	357	8	573	-	-
3	-	539	537	9	551	-	-
4	-	593	614	10	561	-	-
5	-	508	491	11	642	-	-
6	-	376	-				-

*Manual count has been adjusted for area over which count was performed to aid comparison

Table 4.8: *Averages and standard deviations for different counting methods*

	Automated thick coverslip	Automated thin coverslip	Manual thin coverslip
Average	589	466	521
Std Deviation	39	116	105

The results clearly show that the use of a thin coverslip introduces added variability in the counting procedure. The volume of the counting chamber in which counting occurs also decreases with the use of a thin coverslip. This indicates that in any automatic counting procedure the use of a thick coverslip should give more accurate and reproducible counts. For the purposes of this project, the need to compare automatic and manual counts for validation has resulted in the use of a thin coverslip as the default.

4.1.7 Hough Transform Results

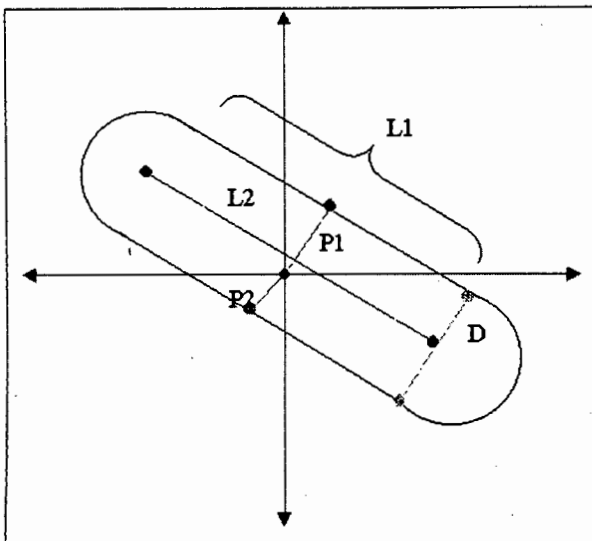
Figure 4.14 shows how bacterial width and breadth measurements were calculated based on the equations for circles and lines as found using the Hough transform. Reference to Figure 4.15 will clarify the explanation given.

Basing cell identification on the presence of parallel lines :

- Cell length is approximated as the largest number of pixels in either of the 2 original lines (each pixel contributes 1 "vote" to the line) plus the breadth between the lines.
- Cell breadth is determined as being the sum of the p–values as these are perpendicular to the lines in the original image and pass through the origin

Basing cell identification on the presence of circles :

- Cell length is approximated as the sum of the distance between the circle centres and the diameter of a circle.
- Cell breadth is approximated as the diameter of a circle.



Length based on lines = L1 + P1 + P2; where L1 is the length based on the longer of the 2 paralell lines and the sum of P1 and P2 is the breadth (assumed to equal the diameter of the end circles);

Breadth based on lines = P1 + P2

Length based on circles = L2 + D, where L2 is the distance between circle centres

Breadth based on circles = D

Figure 4.14: Length and breadth measurement for the Hough transform application

Time constraints on the processing of images resulted in only a few cells being processed by the Hough transform algorithm, developed using a Visual Basic program (See attached CD-ROM and Appendix J). Cells processed using this method are shown in Figure 4.15. Their length, breadth and area measurements as calculated by Optimas as well as their Hough transform characteristic equations are indicated. The Optimas measurements are compared to the length and breadth measurements deduced using the Hough transform. The processing time to extract data for the Optimas system is negligible (less than 1 second) while that to process each cell in the Visual basic program is listed in Figure 4.15. Figure 4.15 also reports the number of white pixels in each cellular image as the processing time is dependent on this number.

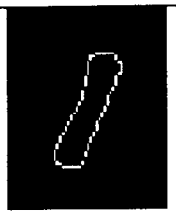
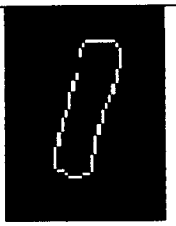
Image (enlarged and not to scale)			
Number of Pixels (white)	100	94	80
Time to process for Hough transform	2 sec for lines, 2 min 13 sec for circles	2 sec for lines 1 min 44 sec for circles	1.5 sec for lines 1 min 11 sec for circles
Optimas Measurement	Area = 589.9 pixels Length = 49.4 pixels Breadth = 15.5 pixels Rectangular Breadth* = 11.9 pixels	Area = 504.0 pixels Length = 46.5 pixels Breadth = 15.5 pixels Rectangular Breadth* = 10.8 pixels	Area = 469.0 pixels Length = 41.3 pixels Breadth = 15.5 pixels Rectangular Breadth* = 11.4 pixels
Hough Transform Measurement based on parallel lines (See calculation method figure 4.14)	30 votes at $\theta = 81$, $p = 4$ and 23 votes at $\theta = 80$, $p = -7$; Therefore length = 41pixels, breadth = 11 pixels.	31 votes at $\theta = 157$, $p = -5$ and 19 votes at $\theta = 156$, $p = 4$; Therefore length = 40 pixels, breadth = 9 pixels.	24 votes at $\theta = 165$, $p = 6$ and 23 votes at $\theta = 161$, $p = -4$; Therefore length = 34 pixels, breadth = 10 pixels.
Hough Transform Measurement on Circles (x,y,r)	$(x,y,r)_1 = 49,27,6$ $(x,y,r)_2 = 15,22,6$ Therefore length = 46 pixels, breadth = 12 pixels	$(x,y,r)_1 = 22,51,6$ $(x,y,r)_2 = 33,22,7$ Therefore length = 44 pixels, breadth = 13 pixels	$(x,y,r)_1 = 18,40,5$ $(x,y,r)_2 = 26,13,5$ Therefore length = 38 pixels, breadth = 10 pixels

Figure 4.15: *Hough Transform results compared to Optimas. Note that images have been enlarged (* Rectangular breadth is calculated as Area/Length)*

The Hough transform results show that the mean width and length is overestimated by Optimas. The overestimation is approximately 10% for the reported length, and 10-50% for the reported width. If the width is approximated by dividing the reported length into the reported area (i.e. the rectangular breadth), then closer values are obtained (0-20% variation). This rectangular breadth is thus assumed more accurate (Figure 4.15).

4.2 BACTERIAL MORPHOLOGY

While the primary objective of this thesis is to establish appropriate image acquisition, processing and analysis techniques for the study of *B. thuringiensis*, the long term goals centres on the correlation of cell morphology and physiological status. To this end, bacterial dimensions, mean internal brightness and cell inclusions are considered during the growth cycle of *B. thuringiensis* to determine whether trends in this data became apparent during processing. Analysis of the mean internal cell brightness was included as during an experimental run it was observed that the cell's internal brightness increased at the onset of the stationary phase of growth before any spore formation became obvious. Since this might give an added insight into the stage of growth and the physiological status of *B. thuringiensis,* the mean internal cell brightness was analysed.

4.2.1 Cell Length

Cell length reached a maximum during logarithmic growth (usually 2-5 hours after innoculation) for all experiments conducted. In shakeflasks, typical cell lengths of 6 μm were found, while maximum cell lengths corresponding to exponential growth of 8 μm were found. The cell length measured in the bioreactor studies was considerably shorter than that in the shake-flask trials. In the bioreactor, cell length decreased from 4.2 to 3.4 μm with progression from the exponential to the stationary phase of growth. This may be due to the increased shear stresses in the highly agitated bioreactor. Under these conditions a shorter cell would be more likely to survive as it would be subject to less extreme force differentials across its length. The determined cell lengths for the bioreactor trial compare favourably with the lengths reported in the literature. Claus and Berkeley (1994) report lengths of between 3 and 5 microns for *B. thuringiensis.* Their measurement method and the conditions under which the bacteria were cultured are not, however, reported. Figures 4.16 and 4.17 show variation in cell length with time on growth in a 2 litre shake flask and the 7 litre bioreactor respectively. Further data on variations in length for all trials undertaken is detailed in Appendix K.

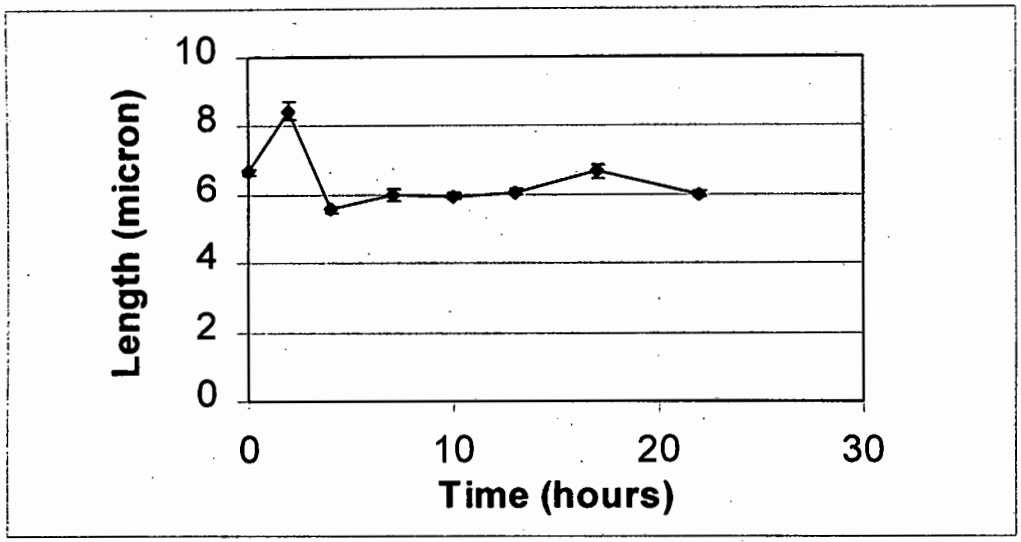


Figure 4.16: *Mean cell length vs time for 2 litre shake-flask, fermentation media*

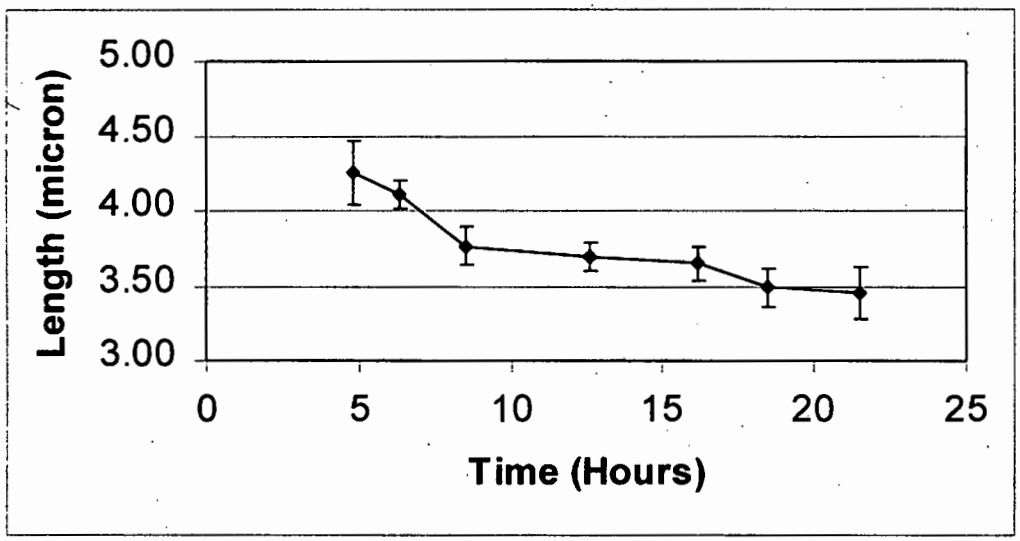


Figure 4.17: *Mean cell length vs time for bioreactor trial, fermentation media .*

In these preliminary trials, the mean cell lengths observed varied on use of the bioreactor versus the shakeflasks, and on use of fermentation media versus nutrient broth (Appendix K). Cell length was markedly shorter for the bioreactor trial, indicating that the cells respond to increasing shear stresses in their environment and adapt accordingly. For the nutrient broth trial in a 1 litre shake flask the cells were shorter during stationary phase (5 μm) than the corresponding fermentation media

trail in a 1 litre shake flask (6 μm), but were longer in logarithmic growth (9μm) than the corresponding fermentation media trial (8 μm). This indicates that media could play a role in bacterial cell length. No differences were observed for different sized shake flasks (Appendix K). It would be useful to conduct a rigorous trial using the robust method developed of measuring cells length to confirm and expand this data.

4.2.2 Cell Breadth

Cell breadth determination based on the method employed by Optimas results in overestimation of the true breadth (Section 3.5.7.4 and 4.1.7). An alternative breadth measurement is the breadth calculated assuming the bacteria are rectangular in shape (i.e. the breadth calculated by dividing the length into the area). This is indicated on Figures 4.18 and 4.19 for comparison.

In Figures 4.18 and 4.19 the variation in bacterial breadth is detailed as a function of growth time for a shakeflask culture (2 litre vessel, fermentation media) and for the bioreactor trial respectively. Both breadths determined by the Optimas function and "rectangular breadth" are reported. No major trends in bacterial breadth in terms of growth rate were observed, while the rectangular breadths are within the range of 1-1.2 μm reported by Claus and Berkeley (1994).

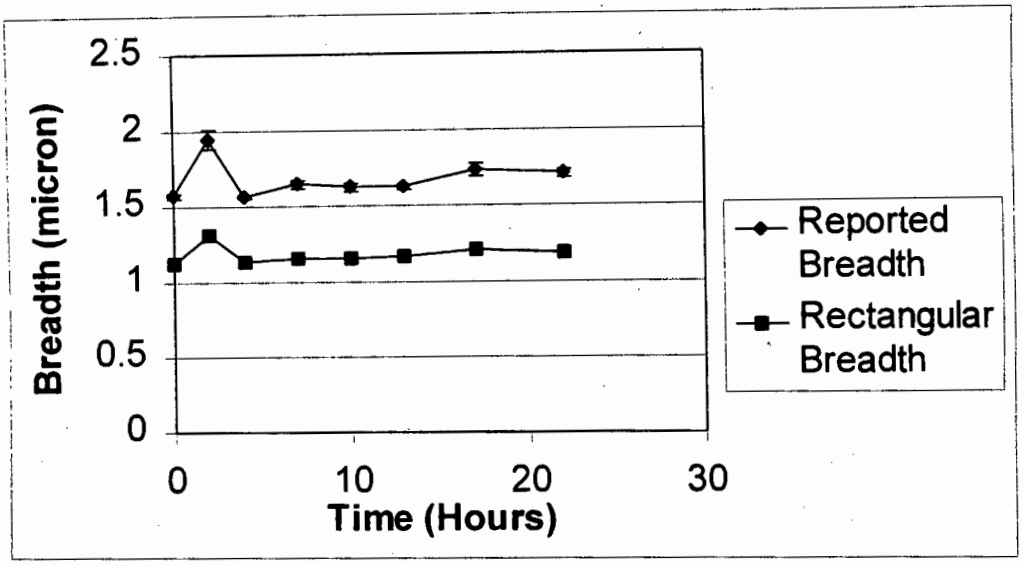


Figure 4.18: *Mean cell breadth vs time for 2 litre shake-flask, fermentation media*

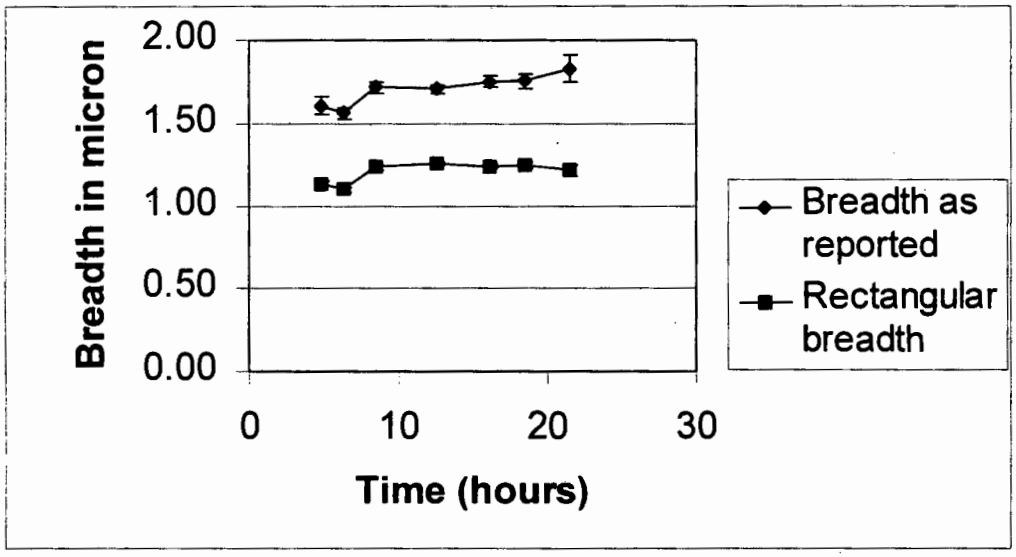


Figure 4.19: *Mean cell breadth vs time for bioreactor trial, fermentation media*

4.2.3 Cell Area

Cell area followed cell length through all trials undertaken, largely because the breadth remained almost constant. Cell area was thus a function only of cell length. Appendix N details the trends of cell area through some of the trials undertaken.

4.2.4 Cell Internal Brightness

Morphology studies showed that as the bacteria approached stationary phase, they exhibited brighter internal regions than during mid-logarithmic growth. This was prior to the formation of any spores or protein crystals. In Figures 4.20 and 4.21 this trend is illustrated for *B.t.* grown on fermentation media in the 7 litre chemap bioreactor under a controlled environment and the 2 litre shakeflask respectively. It is possible that future work may be able to correlate this increase in internal brightness with a physiological property of the bacteria. For instance, correlation of this event to the onset of sporulation may provide a signal for an alteration of culture conditions to maximise insecticidal protein formation in place of growth.

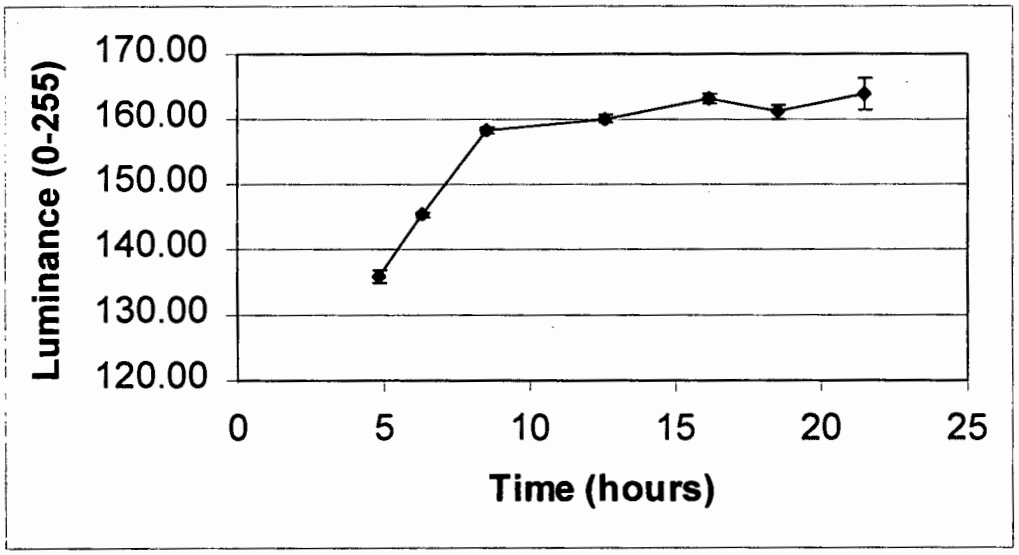


Figure 4.20: *Mean cell internal luminance vs time, bioreactor trial*

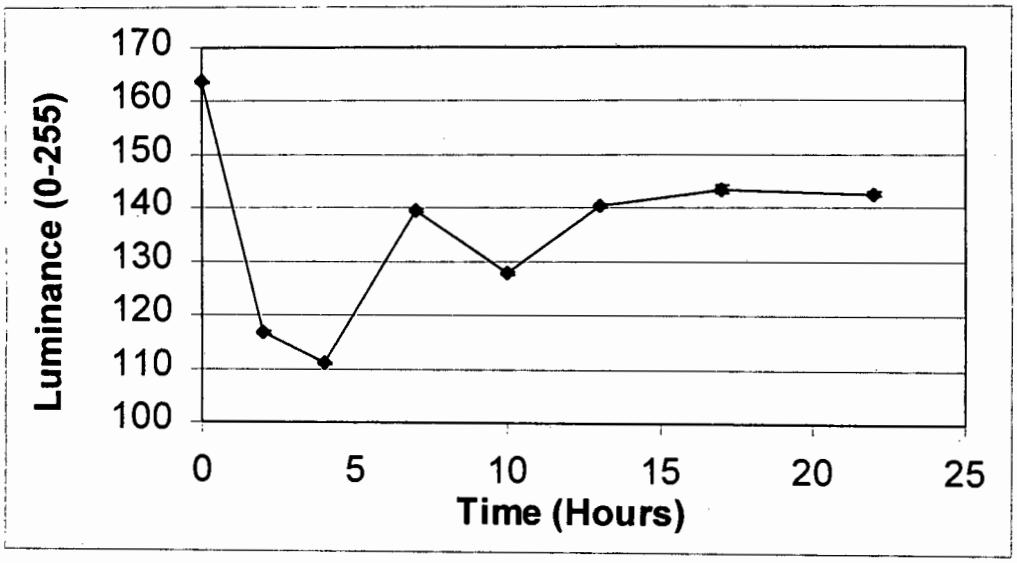


Figure 4.21: *Mean cell internal luminance vs time, large shake-flask*

4.2.5 Sporulation and Protein Production

As *B. thuringiensis* is used as a biological protein producer it is appropriate to report some results on the ratio of spores or proteins produced as a proportion of the total number of cells. Figure 4.22 shows the increase in the percentage of spores and protein crystals found for the bioreactor trial. The percentage of spores and protein crystals present was calculated by dividing the number of spores and proteins in the image by the total number of cells in the image. As the final measurement for this trial was after cell lysis and free spores were counted the percentage sporulation exceeded 100%. Please also refer to Appendix P for the results obtained for the shake-flask trials.

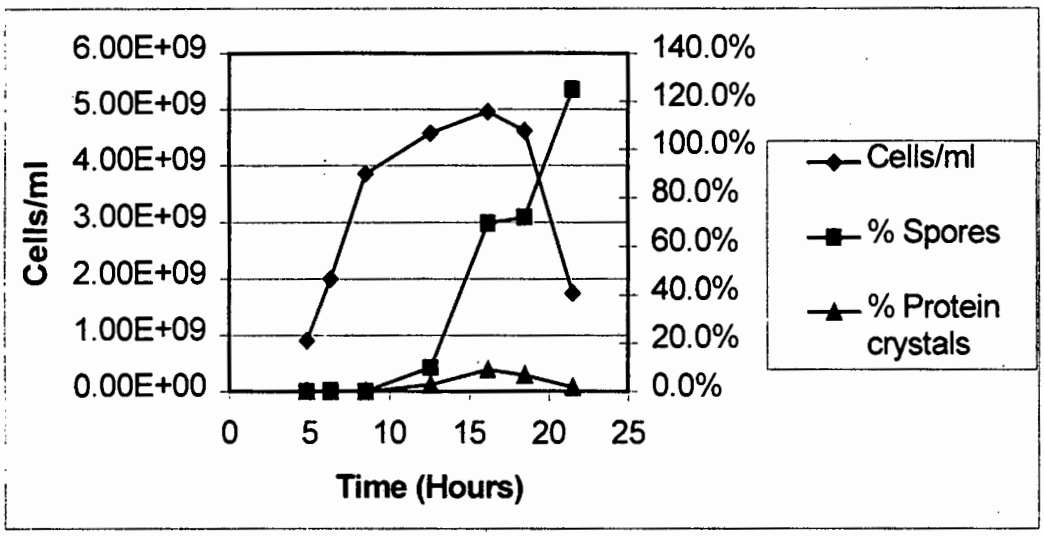


Figure 4.22: *% Spores and Proteins produced during the bioreactor trial. Free spores are also counted which brings the final spore count to over 100%*

The percentage sporulation results compare favourably to the percentage sporulation reported by Holmberg *et al.* (1980). These authors reported approximately 40% sporulation 14 hours after the onset of sporulation at a cell concentration of 2×10^9 cells/ml (Figure 2.1). These similarities were evident despite the fact that that these authors used a different subspecies of *Bacillus thuringiensis* and different fermentation media (Holmberg *et al.*, 1980).

4.3 MOTILITY MEASUREMENTS

To ascertain if motility measurements could be found and, if so, to give an indication of possible methods of accurately determining bacterial motility, a measure of the motility of *B. thurinigiensis* was obtained for a single sample. To do this, two approaches were taken. In the first, a cell was tracked individually through a series of images and its average velocity calculated. In the second, the difference between two successive images was recorded to determine what proportion of cells had moved.

In both approaches a series of 10 images were taken at 0.72 second intervals (the shortest possible time using the frame grabber available). These were then used either to track the movements of individual cells, or to calculate the fraction of cells to have moved between successive frames. Two different series of images, at 127 seconds apart, were acquired from the same sample.

4.3.1 Individual Method

In the individual method, each cell tracked is marked in each image and the x,y co-ordinates of the marked points are extracted. Example data for the tracking of 8 cells in each of 2 series of images is detailed in Appendix O (Tables O.1 and O.2). The distance the bacterium has moved between frames is then calculated. In this way the average velocity of the bacterium over the whole series can be calculated. Figure 4.22 shows the starting position of one of the tracked bacteria (+) as well as the line tracking its motion between successive frames. Note that only the first frame is shown, with the line superimposed on it. Please refer to the web page on the attached CD to view the image sequences.

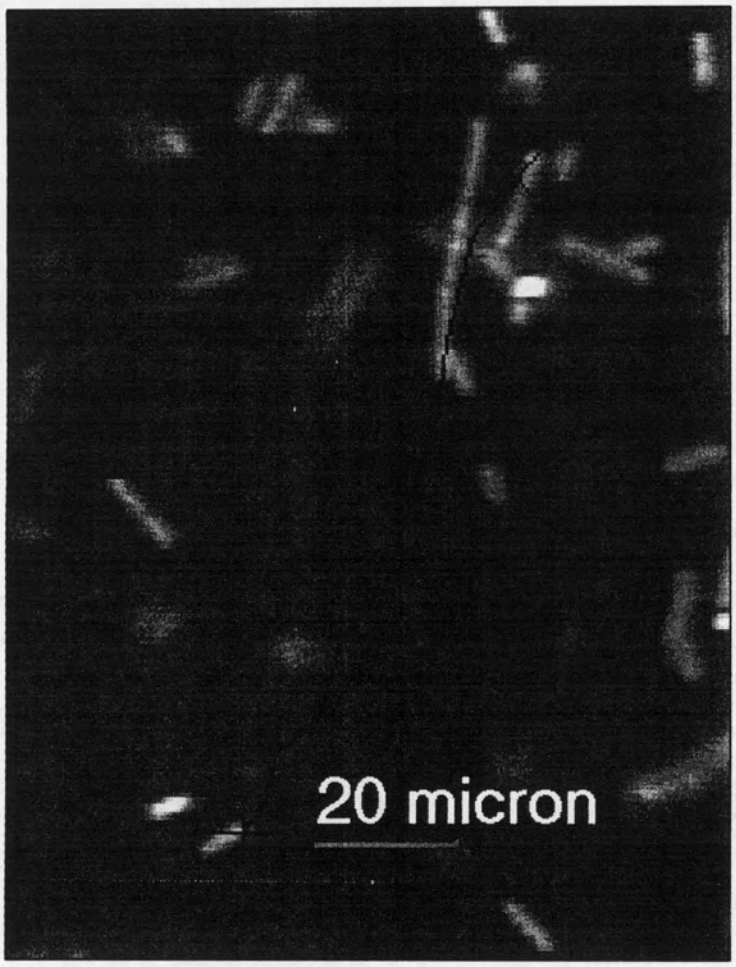


Figure 4.23: *Individual motion of a bacterium*

The individual method permits measurement of the average velocity of any motile bacteria in an image. The method is dependent on the user choosing a representative sample of bacteria. This is not easily achieved as few bacteria that remain in focus in all 10 images in a sequence.

Table 4.9 shows the results obtained for the measurement of individual bacterial motility. It indicates that the motile cells in the counting chamber do not "slow down" with time spent in the counting chamber. The second sequence of 10 images taken 127 seconds after the first shows no significant change in the mean velocity of motile cells as compared to the first sequence.

Table 4.9: *Mean velocity of motile cells determined by the individual method*

Sequence Number	Number of cells analysed	Mean velocity	Error in mean velocity
1	8	13.4 μm/s	±1.2 μm/s
2	8	12.8 μm/s	±1.7 μm/s

4.3.2 Frame Based Method

By analysing the difference in position of all cells between two successive frames, the motility index can be calculated. This motility index is defined as the fraction of the total area of cells in the original image that has moved in the subsequent image (Section 3.6.2.2). A motility index of 100% thus indicates that all cells have moved by at least their own length in the time between successive frames.

Provided the time between image frames in a sequence is constant it is possible to compare the motility indices of different image sequences, i.e. the relative number of motile cells in an image. This could be an indicator of the number of cells in an image which have not become attached to the glass of the counting chamber, and are thus motile, if it is assumed that all unattached cells are motile.

Motility indices for the 2 sets of image sequences were extracted (Appendix O) and are reported in Table 4.10. It was observed that the number of motile cells decreased with time in the counting chamber. For *B. thuringiensis* it is postulated that the cells should gradually become attached to the glass of the counting chamber at a rate proportional to the number of free cells available.

Table 4.10: *Motility indices for images sequences of the same sample at different times*

Sequence	Motility index	Standard Error in Motility index
1	65.6%	2.5%
2 (127 seconds later)	33.0%	1.8%

It is postulated that the rate of attachment of bacteria to the glass of the counting chamber should be influenced by the surface properties of the bacteria. Thus tracking the variation in this rate by using the motility index holds potential in quantifying the adhesion properties of *B. thuringiensis* at different stages of growth. These adhesion properties may be related to the production of exotoxin or indeed, to any number of physiological changes in the bacterium.

5 CONCLUSIONS AND RECOMMENDATIONS

5.1 INTRODUCTION

The objectives of this study were to develop robust methods for the quantification of the morphology of the bacterium, *Bacillus thuringiensis,* the identification of intracellular objects in the bacterium and the automatic counting of the bacterium. The quantification of the morphology of the bacterium and identification of the presence of intracellular spores and/or protein crystals is potentially beneficial for the control of large scale fermentations and in increasing the understanding of which different morphological indicators correspond to which physiological changes in the bacteria. The automatic counting of bacteria is potentially more accurate than manual counting while feedback to an industrial process is much faster than other measurements of bacterial growth such as dry mass measurements.

5.2 IMAGE ACQUISITION AND PROCESSING

Image analysis of the morphology of a single celled organism in a controlled lighting environment (a microscope) provides some challenges to extracting fast and reliable data. In this thesis, methods to automatically image analyse *Bacillus thuringiensis* accurately and reliably have been developed. These have addressed the following:

- Ensuring that all bacteria in an image are in focus and are orientated correctly
- Storing images such that any image compression employed does not significantly alter the data obtained
- Evaluating the number of images required in order to give statistically meaningful results
- Preventing the presence of intracellular spores and protein crystals from influencing the location of bacteria and of bacterial boundaries.
- Adequately separating overlapping bacteria
- Accounting for inaccuracies in computer determined dimensions
- Accurately locating spores and intracellular protein crystals

5.2.1 Sample Preparation

The method by which the bacterial sample to be viewed is mounted on water agar has proved to be effective in increasing the accuracy of locating the spores and protein crystals in the bacteria. It did not prove to have any effect when evaluating the number and accuracy of locating the bacteria themselves. This was a result of the robustness of the image analysis procedures developed as out of focus cells are generally eliminated by these procedures. The water agar method has proven to work for other bacterial samples, notably *Corynebacterium glutamicum* (Henwood, Personal Communication), and shows potential as a general preparation technique for all bacterial image analysis applications.

It is recommended that wet, agar mounted cells be photographed under phase contrast microscopy to generate the best images for later processing.

5.2.2 Image Storage

The storage of images as highly compressed JPEG images in which some loss of image quality occurred did not significantly alter the data obtained in the study of *B. thuringiensis*. Thus using the JPEG compression system for these specific images under the image processing routines specified is acceptable. Nevertheless, the storage of images for possible future work, using unknown routines, requires that the images be stored in as high a quality format as possible. It is not recommended that the JPEG image compression method be used unless there is no alternative. The hardware solution of writing all images to CD as TIFF files is recommended.

5.2.3 Number of Images

Thirty images per fermentation sample was shown to give accurate results and sufficiently low variability. The standard error in the reported cell length decreased from 1.33 μm to 0.54 μm when the number of images analysed increased from 10 to 30, while total processing time increased from 20 to 60 minutes. An increase in the

number of images from 30 to 90 showed a further decrease to 0.33 μm in the standard error reported. It is concluded that this is an insufficient narrowing of the error given the increased processing time and storage requirements; hence 30 images is adequate.

5.2.4 Image Processing

Manual validation of the fully automated procedure showed that cells were accurately identified at a level above 90% at all times. Thus all problems associated with spores and protein crystal inclusions, overlapping cells and background irregularities were adequately handled by the procedures developed.

The use of templates as markers for the identification of spores and proteins and for image subtractions proved effective at overcoming any edge effects caused by the presence of an intracellular protein crystal or spore. The templates employed also proved useful in accurately ascertaining the inclusion of objects within bacteria and thus on evaluating whether a found bright object is a protein crystal or not. This was based on evaluating whether a spore was also included in the same bacteria. In the event of this image processing routine being employed for another bacteria in which protein crystal production is not sporulation dependent, the identification procedure for finding protein crystals would need to be changed.

The watershed separate algorithm as applied to the inverted distance transform of the binary image is very effective at separating touching bacterial cells. This is because the surface irregularities on the perimeter of the cells are of an order of magnitude lower than the concavities at the joining of the cells. Thus, provided the watershed preflood is set in the correct range, the cells are accurately separated without the problems of under or over separation associated with such procedures as the use of ultimate erosion markers.

Background irregularities did not significantly affect data obtained as they were easily eliminated based on size or shape criteria before the construction of the templates for protein crystal, spore and bacterial identification.

5.2.5 Measurement Extraction

Measurements extracted by the Optimas software system needed to be properly interpreted. Length and breadth measurements were actually the major axis length and the sum of the largest distances from the major axis length to the bacterial perimeter, respectively. This is not the same as the length and breadth a human operator would measure. To correct this, two approaches were taken. The first was the calculation of a "rectangular breadth" which calculated the bacterial breadth by approximating the bacterial cell as a rectangle and dividing its area by the reported length. The other approach was to describe the shape of the bacteria as two circles joined by parallel lines and then use a shape recognition algorithm, such as the Hough transform, to extract length and breadth measurements.

The first approach assumed that the reported length and area were correct. This was shown to be a reasonable assumption based on the Optimas measurement of length and area for the characteristic bacterial shape. The "rectangular" breadth and reported lengths were within 10% of the breadths and lengths as calculated by the Hough transform.

The use of the Hough transform as a method to extract accurate determinations of bacterial shape shows some promise. The computationally expensive algorithm developed could be speeded up by improving the program coding and using more advanced computers. It is envisaged that such an approach holds the key to the best method of quantifying bacterial shape and accurately extracting bacterial dimensions.

5.3 BACTERIAL MORPHOLOGY

The results obtained for bacterial length, breadth, area and internal luminosity showed definite variation with bacterial growth stage, the type of media in which growth had occurred and the external shear environment.

Bacterial length showed the most marked change with growth stage. Rapid growth showed longer bacterial cells, with maximum cell lengths of 8 to 9 μm for the shake flask trials and 4.5 μm for the bioreactor trial. These dropped to 5 to 6 μm and 3.5 μm for the stationary phase respectively. The difference in measured bacterial cell length for the shake flask trials and that for the bioreactor trial show the distinct influence of shear on the maximum bacterial lengths obtainable. Literature reported lengths were 3 to 5 μm.

Bacterial breadth remained constant at a rectangular breadth of 1.1 to 1.2 μm for all trails undertaken. This agrees with literature reported figures. Bacterial area was thus only a function of bacterial length.

Internal cell brightness showed a distinct increase during late logarithmic growth, before the appearance of any intracellular spores or protein crystals. This is possibly a result of internal changes in the bacterial cell. This phenomenon could be used for future fermentation control strategies in which culture conditions which are favourable for sporulation and protein production as opposed to cell growth, are introduced.

Spore and protein production achieved was similar to that reported in the literature. Spores were identified with an accuracy greater than 90%, while the identification of proteins was over 80%. Thus the presence of spores and proteins and their identification using image analysis has been demonstrated.

5.4 AUTOMATIC COUNTING PROCEDURES

Manual counting of bacterial cells was shown to introduce errors when the counts were performed at 1000× magnification. The reason was the proximity of the focal point of the 100× objective lens to the lens itself. This necessitated the use of a thin coverslip, as the objective lens focussed within a thick coverslip. A thin coverslip, however, is prone to distortion and thus introduces inaccuracies and variability in the data obtained (up to 20 % lower counts).

A method to overcome this problem was proposed, based on automatically counting bacterial cells at 200× magnification under dark field illumination. It was shown that the automatic counting of cells at this magnification using a thin coverslip correlated well with manual counts at a 1000× using the same slide preparation. At 200×, however, a thick coverslip may be used. Once it was shown that the manual and automatic counts correlated well, it was further demonstrated that automatic counts using a thick coverslip showed much less variability. It can thus be concluded that a 200× dark field image prepared using a thick coverslip would give a good basis for a reproducible automatic counting procedure provided minimal cell clumping occurs.

5.5 MOTILITY MEASUREMENTS

Individual cell velocity was measured and it was shown that meaningful data is extractable. Each cell was manually tracked through 10 succesive images, acquired at 0.72 second intervals. Care was needed to ensure that the cells did not leave the focal plane during the study. The process is time consuming and labour intensive and is not ideal for automation.

Measurements of bulk motility, or the fraction of cells in an image which are motile, was shown to be easily achieved. A motility index was extracted for each image sequence and it was shown that this decreased with time spent in the counting chamber. This motility index was automatically obtained. It is postulated that for *B. thuringiensis* the motility index is actually a measurement of the fraction of cells which are not attached to the glass surfaces in the image. The comparison of motility index data as a function of time spent in the counting chamber can thus give attachment rate data, which might vary during the course of a fermentation. Other bacterial motility indices will not necessarily reflect the degree of attachment but may give the percentage of cell which actually are motile in an image.

5.6 FURTHER WORK

The accurate image analysis of *B. thuringiensis* has been demonstrated. This facilitates further work in the following fields:

- Developing knowledge as to the relationship between bacterial morphological and physiological changes. This could help predict the onset of sporulation, warn of adverse culture conditions and help to project protein yields early in an industrial process, allowing for better control strategies.
- Fully automating the entire process such that sampling and slide preparation are automated.

In addition, work on bacterial shape quantification has shown promise. This may provide more accurate bacterial dimensions. Work here should centre around improving the processing speed to apply the Hough transform as well as improving the algorithms developed. Integration of a developed procedure into the Optimas environment would then enable more accurate and determination of bacterial dimensions without sacrificing automation.

Bacterial motility studies are another field of possible future research. Bacterial motion has been shown to be measurable and thus changes with changing fermentation and culture conditions should be obtainable.

6 REFERENCES

Adams, H.L. and Thomas, C.R. (1987), "The use of image analysis for morphological measurements on filamentous microorganisms", *Biotechnology and Bioengineering*, **32**: 707-712

Agaisse, H. and Lereclus, D. (1995), "How does *Bacillus thuringiensis* produce so much insecticidal crystal protein?", *Journal of Bacteriology*, **177:** 6027-6032

Arcas, J., Yantorno, O. and Ertola, R. (1987) "Effect of high concentration of nutrients on *Bacillus thuringiensis* cultures", *Biotechnology Letters,* **9**: 105 - 110

Arcas, J., Yantorno, O., Arrarás, E. amd Ertola, R. (1984) "A new medium for the growth and delta-endotoxin production by *Bacillus thuringiensis* var. *kurstaki*", *Biotechnology Letters*, **6**: 495-500

Astola, J. and Dougherty, E.R. (1994) "Nonlinear filters", In: *Digital Image Processing Methods*, E.R. Dougherty, ed., Optical Engineering, 1-42. New York, Basel, Hong Kong: Marcel Decker

Barnett, M. (1992), "Microbiology Laboratory Exercises", *Wm. C. Brown Communications,* Dubuque

Beucher, S. and Meyer, F. (1993), "The Morphological approach to segmentation: the watershed transformation", *Mathematical Morphology in Image Processing,* E.R. Dougherty, ed., Optical Engineering, pp. 433-482. New York, Basel, Hong Kong: Marcel Decker

Bleu, A., De Guise, J. and LeBlanc, A.R. (1991), "A new set of fast algorithms for mathematical morphology", *Computer Vision, Graphics, and Image Understanding,* **56:** 210-229

Borgefors, D. (1986) "Distance transformations in digital images", *Computer Vision, Graphics, and Image Understanding*, **34:** 344-371

Bulla, L.A., Kramer, K.J. and Davidson, L.I. (1977) "Characterisation of the entomocidal parasporal crystal of *Bacillus thuringiensis*", *Journal of Bacteriology,* **130:** 375 - 383

Camper. A., Hayes, J.T., Sturman, P.J., Jones, W.L. and Cunningham, A.B. (1993) "Effects of motility and adsorption rate coefficient on transport of bacteria through saturated pourous media.", *Applied and Environmental Microbiology,* **59**: 3455-3462

Claus, D. and Berkeley, R.C.W. (1994) "*Genus* bacillus", *Bergey's Manual of systematic bacteriology*, **2:** 1105

Costello, P.J. and Monk, P.R. (1985) "Image analysis method for the rapid counting of *Saccharomyces cerevisiae* cell", Applied and Environmental microbiology, **49:** 863-866

Cox, P.W. and Thomas, C.R. (1992) "Classification and measurement of fungal pellets by automated image analysis", *Biotechnology and Bioengineering*, **39:** 845 - 952

Dabee, S. (1996) "Maximising Oxygen Transfer and Minimising Shear Damage in the Microbial Production of GLA by *Mucor Rouxii*", Masters Thesis Dissertation, Chemical Engineering Department, University of Cape Town

Donachie, W.D., Begg, K.J. and Vicente, M. (1976) "Cell length, cell growth and cell division.", *Nature*, **264**: 328-333

Dougherty, E.R. (1994) "Introduction", *Digital Image Processing Methods*, E.R. Dougherty, ed., Optical Engineering, xi-xxvi. New York, Basel, Hong Kong: Marcel Decker

Gonzalez, R.C. and Woods, R.E. (1992) "Digital image processing", *Addison-Wesley Publishing Company*, New York

Heijmans, H.J.A.M. (1995) "Mathematical morphology: a modern approach in image processing based on algebra and geometry.", *SIAM Review*, **37**: 1-36

Holmberg, A., Sievänen, R. and Carlberg, G. (1980) " Fermentation of *Bacillus thuringiensis* for exotoxin production: process analysis study", *Biotechnology and Bioengineering*, **22:** 1707 -1724

Huls, P.G., Nanninga, N., van Spronson, E.A., Valkneburg, J.A.C., Vischer, N.O.E., Woldringh, C.L. (1992) "A computer-aided measuring system for the characterisation of

yeast populations combining 2D - image analysis, electronic particle counter and flowcytometry." *Biotechnology and Bioengineering,* **39:** 343 - 350

James, G.A., Korber, D.R., Cladwell, D.E. and Costerton, J.W. (1995) " Digital image analysis of growth and starvation responses of a surface-colonizing *Acinetobacter* sp.", *Journal of Bateriology*, **177:** 907-915

Kieran , P.M., O'Donnell, H.J., Malone, D.M. and MacLouglin, P.F. (1994) "Fluid shear effects on suspension cultures of *Morinda citrifolia* ", *Biotechnology and Bioengineering*, **45:** 415 - 425

Lawrence, J.R., Korber, D.R. and Caldwell, D.E. (1992) "Behavioural analysis of *Vibrio parahaemolyticus* variants in high- and low-viscosity microenvironments by use of digital image processing.", *Journal of Bacteriology,* **174**: 5732-5739

Leavers, V.F. (1992a) "Survey: which Hough transform?", *Computer Vision, Graphics, and Image Understanding,* **58**: 250-264

Leavers, V.F. (1992b) "Use of the Radon transform as a method of extracting information about shape in two dimensions.", *Image and Vision Computing*, **10**: 99-107

Leavers, V.F. (1994) "Preattentive computer vision: towards a two-stage computer vision system for the extraction of qualitative descriptors and the cues for focus of attention.", *Image and Vision Computing*, **12**: 583-599

Meadows, M.P., Ellis, D.J., Butt, J., Jarret, P., Burges, H.D. (1992) "Distribution, frequency and diversity of *Bacillus thuringiensis* in an animal feed mill", *Applied and environmental microbiology*, **58:** 1344-1350

Miller, I., Freund, J.E. and Johnson, R.A. (1990) "Probability and Statistics for Engineers", Prentice Hall International Publishing, London

Moga, A.N. and Gabbouj, M. (1997) "Parallel image component labelling with watershed transformation.", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **19:** 441 - 450

Najman, L. and Schmitt, M. (1996) " Geodesic saliency of watershed contours and hierachical segmentation.", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* **18:** 1163 - 1173

Nielsen, J. and Krabben, P. (1995) "Hyphal growth and fragmentation of *penicillium chrysogenum* in submerged cultures", *Biotechnology and Bioengineering*, **46**: 588-598

Noble, J.A. (1996) "The effect of morphological filters on texture boundary localisation.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**: 554-561

O'Shea, D.G. and Walsh, P.K. (1996) "Morphological characterisationn of the dimorphic yeast *Kluyveromyces marxianus* var. *marxianus* NRRLy2415 by semi-automated image analysis", *Biotechnology and Bioengineering*, **51:** 679-690

Olsvik, E.,Tucker, K.G., Thomas, C.R. and Kristiansen, B. (1993) " Correlation of *Aspergillus niger* broth reological properties with biomass concentration and the shape of mycelial aggregates.", *Biotechnology and Bioengineering*, **42**: 1046-1052

Optimas (1994a) "Optimas users guide volume 1", Optimas Corporation

Optimas (1994b) "Optimas users guide volume 2", Optimas Corporation

Otsu, N. (1979) "A threshold selection method from gray-level histograms.", *IEEE Transactions on systems, management and cybernetics*, **SMC-9**:62-66

Packer, H.L. and Thomas, C.R. (1989) "Morphological measurements on filamentous microorganisms by fully automatic image analysis", *Biotechnology and Bioengineering,* **35:** 870-881

Packer, H.L., Keshavaraz-Moore, E. and Lilly, M.D. (1991) "Estimation of cell volume and biomass of *Penicillium chrysogenum* using image analysis", *Biotechnology and Bioengineering*, **39**: 384-391

Paul, G.C., Kent, C.A. and Thomas, C.R. (1992) "Viability testing and characterisation of germination of fungal spores by automatic image analysis", *Biotechnology and Bioengineering* **42:**11-23

Phillips, B.R., Quinn, J.A. and Goldfine, H. (1994) "Random motility of swimming bacteria: single cells compared to cell population.", *American Institute of Chemical Engineers Journal*, **40**: 334-348

Pons, M.N., Vivier, H., Remy, J.F. and Dodds, J.A. (1993) " Morphological characterisation of yeast by image analysis", *Biotechnology and Bioengineering*, **42:** 1352- 1359

Poole, P. (1990) "Microbes on the move", *New Scientist*, **March 3** : 38-41

Reichl, U., King, R. and Gilles, E.D. (1992) "Characterisation of pellet morphology during submerged growth of *Streptomyces tendae* by image analysis.", *Biotechnology and Bioengineering*, **39**: 164-170

Rosenfeld, A. (1989) "Image Analysis", In: *Digital Image Processing Techniques*, M.P. Ekstrom, ed., Academic Press Inc, 257-287, New York, London, Sydney, Tokyo

Sambrook, J., Fritsch, E.F., and Maniatis, T. (1989) "Molecular Cloning. A laboratory manual." Second Edition. Cold Spring Harbour Laboratory Press, Cold Spring Harbour, New York

Shneier, M. and Abdel-Mottaleb, M. (1996) "Exploiting the JPEG compression scheme for image retrieval.", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **18**: 849-853

Sieracki, M.E., Reichenbach, S.E. and Webb, K.L. (1989) "Evaluation of automated threshold selection methods for accurately sizing microscopic fluorescent cells by image analysis.", *Applied and Environmental Microbiology*, **55**: 2762-2772

Singh, A., Yu, F-P., McFeters, G.A. (1990) " Rapid detection of Chlorine-induced bacterial injury by the direct viable count method using image analysis.", *Applied and Environmental microbiology*, **56:** 389-394

Sneath, P. H. A. (1986) "Endospore-forming Gram-Positive Rods and Cocci" In: Bergey's Manual of Systematic Bacteriology, **2:** 1109

Suhr, H., Wehnert, G., Schneider, K., Bittner, C., Scholz, T., Geissler, P. , Jähne, B. and Scheper, T. (1995) "In situ microscopy for on-line characterisation of cell-populations in

bioreactors, including cell concentration measurements by depth from focus.", *Biotechnology and Bioengineering*, **47**: 106-116

Treskatis, S.-K., Orgeldinger, V., Wolf, H. and Gilles, E.D. (1996) "Morphological characterisation of filamentous microorganisms in submerged cultures by on-line digital image analysis and pattern recognition.", *Biotechnology and Bioengineering,* **53**: 191-201

Tukey, J.W. (1974) "Nonlinear (nonsuperposable) methods for smoothing data", *Congress Recordings, EASCON,* 673. (As cited by Astola, 1994)

Tzeng, Y-M. and Young, Y-U. (1995) "Peniccillin-G enhanced production of thuringiensin by *Bacillus thuringiensis* sp. *darmstadiensis.*", *Biotechnology Progress*, **11**: 231-234

Vaija, J., Lagaude, A. and Ghommidh, C. (1995) "Evaluation of image analysis and laser granulometry for microbial cell sizing", *Antonie van Leeuwenhoek*, **67**: 139-149

Vincent, L. and Soille, P. (1991) " Watersheds in digital spaces: an efficient algorithm based on immersion simulations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **13:** 583 - 598

Vincente, A., Meinders, J.M. and Teixera, J.A. (1996) "Sizing and counting of *sccharomyces cerevisiae* floc populations by image analysis, using an automatically calculated threshold.", *Biotechnology and Bioengineering*, **51**:673-678

Wallace, G. (1992) " The JPEG still picture compression standard", *IEEE Transactions on Consumer Electronics*, **38:** xviii-xxxiv

Weinzerl, R. and Solter, L. (1997) "The insect pathogen *Bacillus thuringiensis*", *Midwest Biological Control News*, as reported at http://www.wisc.edu/entomology/mbcn/fea207.html

Wu, K., Gauthier, D. and Levine, M.D. (1995) "Live cell image segmentation", *IEEE Transactions on Biomedical Engineering*, **42**: 1-12

Yamashita, Y., Kuwashima, M., Nonaka, T. and Suzuki, M. (1993) "On-line measurement of cell size distribution and concentration of yeast by image processing.", *Journal of Chemical Engineering of Japan*, **26**: 615-619

APPENDIX A: FERMENTER OPERATION

In the operation of the fermenter it was necessary to ensure that there was no contamination, while maximising oxygen transfer through increasing the incoming air flow rate and the impeller speed. Listed below please find the fermenter operating procedures.

1. Ensure the fermenter is **clean** and dry. Fit rubber seals on both sides of the glass tube into the fermenter base and lid. Place the 4 bars and nuts in place and tighten progressively moving from one bar to the next in a clockwise fashion.

2. Place the glucose solution into the fermenter.

3. Ensure all ports have rubber seals and are screwed down **tightly**.

4. Calibrate the pH electrode at pH 7.0 at room temperature. This calibration point is independent of temperature. Adjust to pH 5.5 after sterilisation and salt and other media addition to the glucose media. Place pH probe into it's port. The pH probe has a longer shaft than all the others and uses a longer injection port. Ensure that the air needle port is clear. Place air condenser into the centre of the fermenter lid.

5. Ensure the boiler is filled with water to the maximum mark. Turn on the boiler and wait for 1.5 bar of steam to be generated.

6. **Close** the green cooling valve on the fermenter; Open the red steam valve; Close the water reflux valve; **Ensure** that the temperature controller is off; Place stirrer on maximum speed; **Ensure** that the air supply is off; Connect the reflux tubing to ensure fast cooling if necessary.

7. Place the stainless steel shield around the fermenter. Wait until the fermenter reaches 100°C and steam just starts to spit our of the exhaust air group. Close the valve at the top of the air group. If the glucose solution starts to move up the condenser switch off the steam and open the cooling water. Once the glucose solution has receded close the cooling value, open the steam value and start to pressurise.

8. When the fermenter has reached 1.2 bar (guage), time the sterilisation process for 20 minutes. Remember to increase the pressure of the pH probe to the pressure of the fermenter (pressure of fermenter - 0.2 bar [initial pressure of fermenter]). Switch off the steam when the pressure reaches 1.2 bar/121°C. Control the fermenter at this pressure by opening and closing the steam valve but do **not** let the pressure build up too high or drop below 1 bar.

9. After 20 minutes, close the red steam valve and open the green main water valve for cooling fully. Also open the water for the heating coils and the reflux water.

10. Open the water for heating and cooling valve. Once the fermenter pressure reaches 0.5 bar and temperature is less than 100°C, **open and remove the air group valve to prevent a vacuum forming.** Also remember to release the pressure in the pH probe during cooling.

11. Cool to 95°C, close cooling valves and start to place needles for:
    - Air needle; Base control needle; Salts, other media and inoculum transfer line needle

    Unscrew and pull out the plunger in the port.

Ensure that you are wearing **heavy duty boilerman gloves, goggles and a lab coat**. Place 2 ml of 96% alcohol onto the exposed rubber diaphragm and ignite.

Unwrap the needle from the sterilised sleeving and pass the needle through a bunsen burner flame before pushing the needle into the port and screwing the needle down firmly.

Repeat for other injection needles. Continue to cool to 30°C by opening the cooling valve.

13. Place air line onto the fermenter and allow air to slowly pass into the fermenter. Check that bubbles are leaving the base of the sparger at low flow rates and is not leaking by bypassing the air needle sparger seal.

14. Add the salt and other media solutions (autoclaved separately) once the fermenter has reached 30°C.

15. Set the temperature controller to the desired temperature. The fermenter can be left in this state indefinitely - contamination is minimised by the low air flow causing a positive fermenter pressure.

16. When adding the inoculum remember to switch off the air. Also if there is a leak in the base line before disconnecting the line switch off the air. Remember when dealing with the NaOH line that the concentration is 5M so take care and **use gloves**.

17. Set the agitation speed to 900 rpm (450 rpm on dial)

Check the calibration of the pH probe by removing a sample from the fermenter (record the volume) and checking the pH with a calibrated pH meter. Adjust the gain accordingly.

APPENDIX B: IMAGE ACQUISTION

The macro designed to acquire images follows. This macro automates the acquisition and storage of files. The intention is to make the acquisition sufficiently fast so as to minimise variability caused by the user choosing interesting areas of the slide instead of simply choosing random locations to be photographed. X number of photographs are taken at 6 second intervals. The user is warned by a tone that acquisition is about to take place and another tone notifies him/her that the slide may be shifted to a new spot. The 6 second interval between tones allows the user to move the slide and focus, but nothing more. In this way the choice of interesting areas to photograph is reduced.

```
// Macro to take 30 pictures (spaced by tones) and save them as tga colour compressed files.

// Macro saves files as 8-bit RGB TIFF files to path of choice.  The format of the saved files
may be changed by modifying the SaveImage arguments (see OPTIMAS help)

char path = Prompt("Please enter the path to save the data and the file name (including
h)","char");
Integer photnum = Prompt("How many images to acquire?","Integer");
beep(20,1);

integer c=0;
BandToView = 0;
SetColorMode (1 : 1 : 8 : 1 : 3 : 3);
while(c<photnum)
{
beep(30,2);
statusbar = "The photo number is":totext(c);
DelayMS( 6000);
beep(40,2);
Acquire (FALSE);
Freeze ();
SaveImage (path:totext(c),, 8, 4, , "", "", "", TRUE, 0, FALSE, );
c++;
}
beep(45,1);
beep(45,1);
```

APPENDIX C: OBJECT CLASSES

Object classes employed were used to determine whether objects found in an image were in fact the objects that Optimas was looking for. The object classes were thus employed to discriminate between different objects based on some characteristic.

The object classes and their functions are listed in Table C.1.

Table C.1: *Object classes and functions*

Object Class	Applied To	Criteria	Goal
Cells	Possible cells after all processing for cells has been performed (i.e. this is the final check on whether an area is a cell or not.)	Area > 3 μm^2; Std Dev of the Histogram of gray values in cell > 3	Differentiate between small, non-cellular areas; areas which are actually background but are highlighted due to JPEG compression; and real cells
SporeProt	Possible spores and proteins after binarisation of bright areas in an image	Area < 1.5 μm^2; Circularity < 25	Differentiate halo bright regions around cells from real spores and proteins
Spore	Spores after finding areas that are above a threshold luminance of 130	Number of "Child Areas" in an area greater than 0	Only spores should have internal regions of high brightness, therefore should differentiate between spores and proteins
Protein	Possible proteins after all other processing.	Number of "Parent Areas" = 1; Mean Gray Value > 150	Differentiate proteins from other bright areas that are not spores. This is achieved by checking within cells (with already identified spores)
ProtFind	Binarised cell image after subtraction of spores	Number of "Child Areas" > 0	To discriminate between cells with included spores and those without. Once these cells have been identified they can be processed to find proteins.

APPENDIX D: AUTOMATIC COUNTING PROCEDURE

This macro counts the number of bright objects in a dark field image and ascribes these as cells. The macro finds all areas with perimeters greater than 3 pixels.

```
// Define matrix of hours in which sample were taken:
integer fmatrix = prompt("Please enter the hours, separated by colons.","integer");

// Find out how many photo's were taken per image
integer photnum = prompt("How many images were taken per sample?","integer");
char Path = prompt("Please enter path name,eg:C:\\OPTIMAS5\\BT\\octfer22\\cc","char");

// Define Sporecheck, CellCheck and ProtCheck - variables to check whether
spores/cells/proteins have been identified
loopend = VectorLength(fmatrix);
integer count2 = 0;

ActivateMeasurementSet("count200");

while (count2 < loopend)
{
integer count = 0;

while(count < photnum)
{
AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 3.0 : -1.0 : -1.0 : -1.0 : 0.0 : 255.0
: -1.0 : -1.0 : 0.0;
NewImage ("SoftwareFixed", "Greg1", 768:572, 0 : 1 : 8 : 1 : 1 : 1, , ModelImage, );
Calibrate (xCalib);

OpenImage (path:totext(fmatrix[count2]):"h":totext(count),, FALSE, 1);
RunMacro ("C:/OPTIMAS5/dialogs/threshld.mac", );
RunMacro ("C:/OPTIMAS5/dialogs/th2.mac", );
CloseWindow ("Auto Threshold");

CreateArea (, , TRUE);

LinkMeasurementSet(2,"A:/count":totext(fmatrix[count2]):".asc",1,1,FALSE);
MultipleExtract (TRUE);
ExportMeasurementSet ();
CloseWindow("Greg1");
count++;
}
count2++;
}
```

APPENDIX E: MORPHOLOGY PROCESSING

This macro processes all aspects of cellular morphology (including the presence of spores and proteins). The flow diagram for the macro can be found in Figures 3.4 and 3.5. Figure 3.4 details the overall loop structure of the macro (in the program the loops as defined by the count and count2 variables). Figure 3.5 details the processing which occurs within each loop of the macro, including the construction of templates (Greg1 to Greg3). Please note that this macro is commented (commented line preceded by //) to facilitate understanding each step.

```
// Macro Written on Tue 28/10 onwards to process cells completely.
// This macro is fully automated and aims to extract the following:
// Cell number, area, breadth, length, perimeter and average gray scale luminosity (mArGV);
// Spore number, area, breadth, length and circularity
// Protein quantity, area and luminosity
// NOTE that this macro requires that only cells with included spores can contain proteins

// Movement studies are performed elsewhere
// Macro uses the overall.cfg configuration file as well as the cells,spores and proteins
// series of measurement sets. THESE MUST BE OPENED PRIOR TO EXECUTION.


// First Switch off undo feature to speed up operation

Undo(FALSE,);

UndoImagesDepth=0;

// Define matrix of hours in which sample were taken:
integer fmatrix = prompt("Please enter the hours, separated by colons.","integer");

// Find out how many photo's were taken per image
integer photnum = prompt("How many images were taken per sample?","integer");
char Path = prompt("Please enter path name,eg:C:\\OPTIMAS5\\BT\\octfer22\\cc","char");

// Define Sporecheck, CellCheck and ProtCheck - variables to check whether
spores/cells/proteins have been identified
Integer Sporecheck = 0;
Integer CellCheck = 0;
Integer ProtCheck = 0;

// Enter loop to process for each set of images ( Each hour )

loopend = VectorLength(fmatrix);

integer count2 = 0;
```

```
while (count2 < loopend)
{

// Reset counters for links to excel

Integer ProteinNumber = 2;
Integer SporeNumber = 2;
Integer CellNumber = 2;


// Define sub-loop in which samples within each hour are incremented
integer count = 0;
while(count < photnum)
{

// Set object classes to false for cells, spores and proteins
// Cells object class eliminates all areas < 2.6 micron squared., SporProt eliminates areas of
circularity greater than 25,
// Spore eliminates all area's without at least one "Daughter"
// Protein eliminates all areas with either a daughter or parent area
ObjectClass (Cells, FALSE);
ObjectClass (SporProt, FALSE);
ObjectClass (Spore, FALSE);
ObjectClass (Protein, FALSE);
ObjectClass (Protfind, FALSE);

// Reset AreaCNV factors to default:

AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 10.0 : -1.0 : -1.0 : -1.0 : 0.0 :
255.0 : -1.0 : 2.0 : 0.0;

// Now we open a new frame, define it as gregl for arithmetic purposes and calibrate it to
1000x
NewImage ("SoftwareFixed", "Gregl", 768:572, 0 : 1 : 8 : 1 : 1 : 1, , ModelImage, );
Calibrate (micl000x);

// Now Open image and start processing
OpenImage (path:totext(fmatrix[count2]):"h":totext(count), 0.0 : 61.531 ::
78.909 : 0.0, FALSE, 1);

// Apply the filters to find and identify regions of rapid change - these correspond generally
to spores and protein edges
Filters ( Median );
Filters (SharpenHigh );

// Now Autothreshold on the light areas. threshold for regions occupying between 0 and 95% of
the area
RunMacro ("C:/OPTIMAS5/dialogs/threshld.mac", );
RunMacro ("C:/OPTIMAS5/dialogs/thb0.mac", );
CloseWindow ("Auto Threshold");
```

```
// Binarise the image
GrayToBinary ();

// Create area's, activate circularity trial (SporProt) , then extract (only if at least one
area falls into range)
CreateArea (, , TRUE);
ObjectClass (SporProt, TRUE);
MultipleExtract (TRUE);
Protcheck = ArTotalTally;

// Eliminate Area's that do not fall into the circularity range and then binary fill to avoid
all holes

if(Protcheck > 0)

{

RunMacro ("C:/OPTIMAS5/BT/SEPT/COMPLEX1.MAC");

BINB_iIterations = 1;
FillFilter(,FALSE);
}
else
{
Threshold (120:121);
GrayToBinary ();
}

// You now should have spores and proteins. Use this image now as a template to extract data
about spores and proteins.

ObjectClass (SporProt, FALSE);



// Duplicate image and process to eliminate spores touching the ROI ( 1 in to correspond to
the same area as the area for cell extraction )
DuplicateImage ();
AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 10.0 : 1.0 : -1.0 : -1.0 : 0.0 : 255.0
: -1.0 : 2.0 : 0.0;
SelectROI( 0.102881: 61.4237::78.8066: 0.107761 );
CreateArea (, , TRUE);

// Open original image so that can extract Spore data
OpenImage (path:totext(fmatrix[count2]):"h":totext(count), 0.0 : 61.531 ::
78.909 : 0.0, FALSE, 1);

// Find and eliminate Proteins by thresholding and finding area's with a great enough internal
luminosity to be spores - define these as a complex Region Of Interest (ROI)
RunMacro ("C:/OPTIMAS5/macsrc/cxroi/cmplxroi.mac", );
RunMacro ("C:/OPTIMAS5/dialogs/compl.mac", );
```

```
CloseWindow ("Complex Regions of Interest");
Threshold (230.0 : 255.0);

// Before creating areas make sure you switch off the removal of area's touching the ROI as
the ROI itself is the spores
AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 10.0 : -1.0 : -1.0 : -1.0 : 0.0 :
255.0 : -1.0 : 2.0 : 0.0;
CreateArea(, , True);
ImageMask (0x0004);

// Now extract only area's with daughter regions - these should correspond to spores

ObjectClass (Spore, TRUE);


// Activate the correct measurement set for the export of Spore Data
ActivateMeasurementSet("Spore");
LinkMeasurementSet(2,"A:/octrawl/spore":totext(fmatrix[count2]):".asc",1,1,FALSE);

// Extract, update Spore number ( Note that this add's one to prevent loss of data in the
event of 0 Spores or in initial case when headers are sent), sporecheck and send data.
MultipleExtract (TRUE);
SporeNumber = SporeNumber + ArTotalTally + 1;
Sporecheck = ArTotalTally;
ExportMeasurementSet ();

// Switch off spore object class,
ObjectClass (Spore, FALSE);

// Binarise this image to form a spore mask for later processing - but only if some spores
were present

if (Sporecheck > 0)

{

RunMacro ("C:/OPTIMAS5/BT/SEPT/COMPLEX1.MAC");

}

Else

{

Threshold (254.0:255.0);

GrayToBinary ();

}
```

```
// Duplicate image, select full screen so that all Cell processing is done in the same area's
as spores, and import image
SelectFullScreen (0);
DuplicateImage ();
OpenImage (path:totext(fmatrix[count2]):"h":totext(count), 0.0 : 61.531 ::
78.909 : 0.0, FALSE, 1);

// Median Area, then Sharpen High
Filters ( Median );
Filters ( SharpenHigh );

// Now Subtract Spores and proteins from this processed image (Helps for binarising )
ArithmeticOp ("Subtract", "Gregl", 0.0 : 61.531 ::
78.909 : 0.0, , "Clip", FALSE, FALSE);

// Autothreshold at 1:95 on Dark objects: - Note that this forces 1% of the Area to Dark
objects. For Area's without spores this is acceptable as the areas become disparate and are
removed under object classes
RunMacro ("C:/OPTIMAS5/dialogs/threshld.mac", );
RunMacro ("C:/OPTIMAS5/dialogs/thll.mac", );
CloseWindow ("Auto Threshold");

// Now binarise, Dilate, fill and erode the images
GrayToBinary ();
Threshold (127.5 : 255.0);
DilateFilter(,1);
FillFilter(,FALSE);
ErodeFilter(,1);

// We now perform an opening to stop any potential threads from being attached to cells:
ErodeFilter (,1);
DilateFilter (,1);

// Now Find Areas and eliminate Areas which are less than 2.6 square micron - these are too
small for cells
CreateArea(, , True);
MultipleExtract (TRUE);
ObjectClass (Cells, TRUE);
CellCheck = ArTotalTally;

// Complex this (if cells present) and eliminate the areas which do not fit the correct
criteria
if(CellCheck > 0)
{
// Perform a separation of these based on the watershed separation.
// This performs a watershed on the distance algorithm of the image to ensure that
// proper separation occurs. It is superior to, but more time consuming than, the binary
separation // macro as it determines relative "peaks" and "valleys" within an area and thus
separates
// only those area's which show distinctive indentations. ( Note that we are preflooding to
9.0 to avoid unnecessary separations which could split cells)
```

```
WatershedSeparateFilter(,9.0,,);
}

// Eliminate areas touching the Defined region of interest ( 1 pixel in as the erosion would
have distorted this)
// and areas which do not fit into the correct size bracket to qualify as unique cells
AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 10.0 : 1.0 : -1.0 : -1.0 : 0.0 : 255.0
: -1.0 : 2.0 : 0.0;
SelectROI( 0.102881: 61.4237::78.8066: 0.107761 );
CreateArea (, , TRUE);

// Activate correct measurement set and link to ASCII file at correct place
ActivateMeasurementSet("Cells");
LinkMeasurementSet(2,"A:/octrawl/Cells":totext(fmatrix[count2]):".asc",1,1,FALSE);
// Extract, update Cell number ( Note that this add's one to prevent loss of data in the event
of 0 Cells or in initial case when headers are sent) and send data.
// ( Note - must first open original file else the gray values extracted are meaningless)
OpenImage (path:totext(fmatrix[count2]):"h":totext(count), 0.0 : 61.531 ::
78.909 : 0.0, FALSE, 1);
MultipleExtract (TRUE);
CellNumber = CellNumber + ArTotalTally + 1;
CellCheck = ArTotalTally;
ExportMeasurementSet ();
ActivateMeasurementSet("EllipseStats");
LinkMeasurementSet(2,"A:/octrawl/Ellips":totext(fmatrix[count2]):".asc",1,1,FALSE);
MultipleExtract (TRUE);
ExportMeasurementSet ();
ObjectClass (Cells, FALSE);
// We now need to find proteins. These are defined as area's which are identified as being in
the spores/crystals category, but which also reside inside cells. In addition this macro
stipulates that a spore must also reside in the area for the additional bright region to
represent a protein.
// Some free proteins are evident but these are not counted as attempting to identify them
results in the identification of regions on out of focus cells
// Re-Binarise cell image. (If cells present - else present a blank screen)
If(CellCheck > 0)
{
RunMacro ("C:/OPTIMAS5/BT/SEPT/COMPLEX1.MAC");

// Subtract greg2 from this - cells with holes then are cells with spores
ArithmeticOp ("Subtract", "Greg2", 0.0 : 61.531 ::
78.909 : 0.0, , "Clip", FALSE, FALSE);
ObjectClass (Protfind, TRUE);
Threshold(127.5:255.0);
CreateArea (, , TRUE);
MultipleExtract (TRUE);
RunMacro ("C:/OPTIMAS5/BT/SEPT/COMPLEX1.MAC");
ObjectClass (Protfind, FALSE);
}
Else
```

```
{
Threshold(254.0:255.0);
GrayToBinary();
}
// Subtract gregl from this
ArithmeticOp ("Subtract", "Greg1", 0.0 : 61.531 ::
78.909 : 0.0, , "Clip", FALSE, FALSE);

// Add Greg 2 to this:
ArithmeticOp ("Add", "Greg2", 0.0 : 61.531 ::
78.909 : 0.0, , "Clip", FALSE, FALSE);

// Find holes - these should be proteins. ( Only Daughter regions)
ObjectClass (Protein, TRUE);

// Set Area CNV factors such that proteins close to edge are not eliminated ( This is done by
changing AreaCNV factors to find areas on 8-coonected blob)
AreaCNVFactors[0..14] = 0.0 : 1.0 : -1.0 : 64.0 : 0.0 : 10.0 : 1.0 : -1.0 : -1.0 : 0.0 : 255.0
: -1.0 : 2.0 : 0.0;
// relocate ROI and find area's
SelectROI( 0.102881: 61.4237::78.8066: 0.107761 );
CreateArea (, , TRUE);

// Open original image so that can extract protein data (luminosity etc)
OpenImage (path:totext(fmatrix[count2]):"h":totext(count), 0.0 : 61.531 ::
78.909 : 0.0, FALSE, 1);

// Activate the correct measurement set for export and link to the correct page and place on
Excel
SelectROI( 0.102881: 61.4237::78.8066: 0.107761 );
ActivateMeasurementSet("Protein");
LinkMeasurementSet(2,"A:/octraw1/Prot":totext(fmatrix[count2]):".asc",1,1,FALSE);

// Extract, update protein number ( Note that this add's one to prevent loss of data in the
event of 0 proteins or in initial case when headers are sent) and send data.
MultipleExtract (TRUE);
ProteinNumber = ProteinNumber + ArTotalTally + 1;
ExportMeasurementSet ();
ObjectClass (Protein, FALSE);

// Macro has now extracted all data required - all that is left is to close down and loop
Closewindow("Greg3");
Closewindow("Greg2");
Closewindow("Greg1");
Sporecheck = 0;
Cellcheck = 0;
ProtCheck = 0;
count++;
}
count2++;
}
```

APPENDIX F: WATERSHED SEPARATION AND MACRO IMAGE SEQUENCES

F.1: LATE STATIONARY PHASE WHERE SPORES AND PROTEINS ARE EVIDENT.

The cells in Figures F.1 and F.2 were taken from the same image. Table 1 contains a pair of cells with visible protein and spore inclusions. These cells did not present a problem to separate. Figure F.2 contains a pair of cells which required separation using the watershed separation algorithm.

 1	 2	 3	 4	 5	 6
1 Original image	2 Median and sharpen high operators applied to identify regions of rapid change. (spores, crystals and cell edges)	3 Automatic thresholding on light objects and binarising of the image.	4 Elimination of incorrectly shaped area's (not circular enough to be spores and crystals) and hole filling	5 Identification of spores based on internal brightness	6 Original image with median and sharpen high operators applied and spores and proteins subtracted
 7	 8	 9	 1		
7 Image binarised after automatic thresholding, then binary dilated, filled and eroded	8 Elimination of non - cellular areas based on a minimum area	9 Application of watershed separate filter.	10 Identified cells	11 Binarised cells, with proteins subtracted	12 Identified proteins

Figure F.1 : *Illustrated identification of spores, proteins and cells.*

1 	2 	3 	4 	5 	6 
1 Original image	2 Median and sharpen high operators applied to identify regions of rapid change. (spores, crystals and cell edges)	3 Automatic thresholding on light objects and binarising of the image.	4 Elimination of incorrectly shaped area's (not circular enough to be spores and crystals) and hole filling	5 Identification of spores based on internal brightness	6 Original image with median and sharpen high operators applied and spores and proteins subtracted
7 	8 	9 	10 	11 	12 
7 Image binarised after automatic thresholding, then binary dilated, filled and eroded	8 Elimination of non - cellular areas based on a minimum area	9 Application of watershed separate filter.	10 Identified cells	11 Binarised cells, with proteins subtracted	12 Identified proteins

Figure F.2: *Further cell identification and separation.*

F.2: INTERESTING CASE IN MID LOGARITHMIC GROWTH

The separation of cells using the watershed algorithm based on the distance transform. (see Section 3.5.6) required accurate setting of the prefill. A prefill of 9 pixel gray levels was chosen as this corresponded to the best separation without the problem of over separation. Figure F.3 is a series of processed images. This is an unusual picture of a clump of cells and demonstrates the application of the watershed filter to separate these cells.

1 	2 	3 	4 
1 Original image	2 Image processed through steps 2 - 8 of table 2.	3 Application of watershed separate filter. This filter is based on the distance transform.	4 Identified cells

Figure F.3 : *Demonstration of the effect of the watershed separate algorithm.*

In image 3 of Figure F.3 above it is apparent that the small, coccoid, cell in the chain is not separated (If this is indeed a separate cell as *Baccillus thuringiesis* does not exhibit this morphology). This is a consequence of the selected prefill. On application of the distance transform to the above image, the coccoid cellular object had a maximum of 35 gray levels, corresponding to distance from the edge. The joining branch to the larger cell was continuous at 27 pixels. The difference of 8 was thus within the prefill of 9 and the object was thus not separated.

APPENDIX G : USE OF THE SHARPEN HIGH FILTER

This is a 5x5 Optimas standard filter that is used to highlight edges in an image. The kernel is as follows:

-1	-1	-1	-1	-1
-1	1	1	1	-1
-1	1	9	1	-1
-1	1	1	1	-1
-1	-1	-1	-1	-1

Figure G.1: *Kernel for the 5x5 Optimas sharpen high filter*

The filter sum is +1, this means that only edges are enhanced and there is no shift in overall gray scale intensity.

The major edge remains at the same pixel location on the image, but the filter does cause other edge effects parallel to the edge. A transition between 2 gray levels could thus become a transition from gray to white, then white to black, then black to gray. Figure G.1 shows the before and after effect.

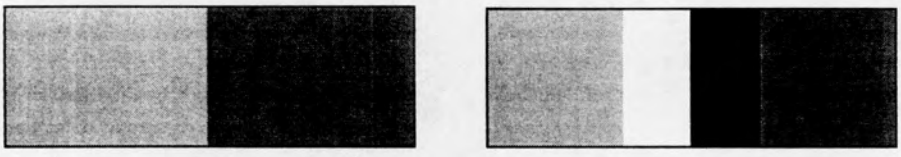


Figure G.2: *Before and after application of the Sharpen High filter.*

The use of the Sharpen High filter for locating cells is justified as the black region formed on the cell boundary is within the cell itself as the cells are darker than the background. After binarisation, binary dilation and binary filling will fill the cells identified from their edges.

APPENDIX H : APPLICATION OF OTSU'S METHOD FOR AUTOMATIC THRESHOLDING

Otsu (1979) proposed a method to automatically segment images based on their histogram of gray scale luminances. This method is employed in Optimas (with slight modifications as in Optimas the objective function may be forced to a certain range). Nowhere, however, is it stated that the method of Otsu (1979) is implemented in Optimas. A spreadsheet application of the method of Otsu's (1979) was thus implemented and the thresholds found were compared to those of Optimas for the same histogram (Figure H.1). The correlation was 100% for non-forced objective functions.

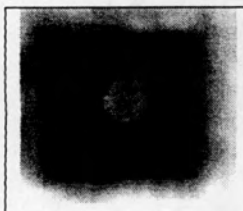
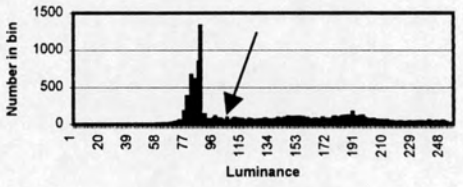
	

Image processed	Histogram of image. – Threshold found for both spreadsheet and Optimas (at 0-255 and 0%-100%) showed threshold at 101. (Marked)

Figure H.1: *Otsu's automatic thresholding*

APPENDIX I: ULTIMATE EROSION METHOD

This is an alternative method to separating joined cells and other objects. It is based on the principle of eroding the object to an ultimate erosion point and from there reconstructing the object with each point corresponding to a separate object. The method works well for smooth objects with well defined centers (Figure I.1) but can easily over-separate objects if they are not sufficiently smooth (i.e. their boundaries are irregular). The Optimas implementation of the ultimate erosion method can limit the number of erosions (i.e. not eroding entirely to a point). This would thus limit over-separation. The problem with this approach is that different cells require different numbers of erosions to be separated (Figure I.2).

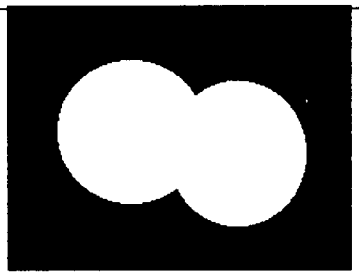
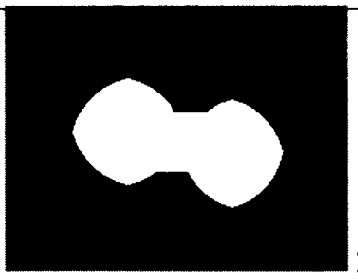
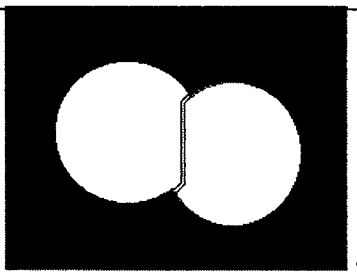
 1	 2
1 Original image – note smooth edges	2 Image after 10 erosions
 3	 4
3 Image after 28 erosions – note that almost "eroded to a point". These points then used as ultimate erosion markers	4 Reconstructed image with division.

Figure I.1: *Application of the ultimate erosion method to objects with smooth edges and well defined centers.*

Figure I.2 demonstrates the use of the ultimate erosion method for a clump of cells. Note how this method is effective at 5 iterations but fails to separate correctly at 4 and overseparates at 6. This causes problems as different cells require different numbers of iterations to achieve adequate separation, thus a global number of iterations for a whole image can not be set, making the automation of such a procedure impossible.

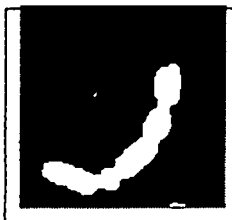
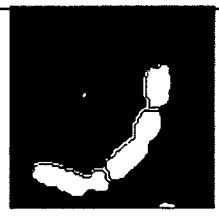
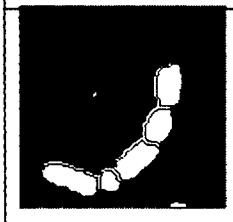
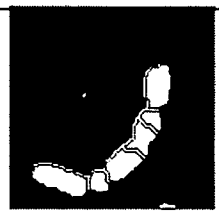
 1	 2
Image of a number of joined cells	Same image after 4 erode iterations to separate cells. Note that the middle cells are not properly separated
 3	 4
Same image after 5 erode iterations to separate cells – separation correct	Same image after 6 erode iterations to separate cells. Note that middle cell now overseparated.

Figure I.2 : *Ultimate erosion separation method for cells.*

Note that the Watershed separation method applied to the same image is detailed in Appendix F.

APPENDIX J: SHAPE RECOGNITION

The implementation of the Hough (or Radon) transform for finding lines and the Hough transform for finding circles in an image is detailed in this Appendix.

Section J.1 gives the program code and operating instructions for the Visual Basic program (See the attached CD for the compiled program). Section J.2 gives the implementation for an Optimas ALI macro. Note that this only finds parallel lines and does not have a facility for finding circles (which the Visual Basic program does).

J.1 PROGRAM CODE AND OPERATING INSTRUCTIONS FOR VISUAL BASIC PROGRAM (ON ATTACHED CD)

J.1.1 Operating Instructions

The attached program first constructs accumulator arrays for the transform spaces of either lines or circles. It then searches through these arrays for maxima, which would correspond to real lines or circles in the original picture. These maxima are then further processed by analysing for unique circles and for unique and parallel sets of lines.

To operate the program choose run on your toolbar. Choose browse and locate the program on the CD (Hough12.exe). Choose OK and the program interface should appear, with an example bacterium in the bacteria window (The paste command under the edit window can replace this bacterium with another picture that has been copied to the windows clipboard). If the program window is too large please adjust your windows settings for screen size under the windows properties settings (right click on the mouse, choose properties – then settings and then choose a screen size of 1280 × 1024 under the desktop area.).

Select the "Shapes" menu and click on "Hough Transform – lines" this will build a line accumulator matrix of the image in the bacterial window. While the operation is proceding the mouse arrow will be displayed as an egg timer.

Now select "Hough Transform – circles". This will build an accumulator array of circles from the bacterial image. This operation should take some time (about 1-2 minutes).

The horizontal scroll bars above the "Num Thresh" radio button can be adjusted to show the threshold of the lines generated. Select a threshold of "Low = 5 High = 100" and press the "Thresh L" button. A threshold with 2 curves should appear in the line threshold window. This is described in detail by Leavers (1992b). If the lower value is adjusted upwards the curves should resolve into 4 points. These represent the 2 lines (each point is at either 0 or 180 degrees, which is mathematically the same line).

The Hough transform for circles constructs a three dimensional array and as such cannot be visually shown here. Nevertheless the 2nd order Radon transform has been coded. Select "Radon – circles") under the shapes menu. This will construct a radon transform for circles based on the threshold chosen for lines. It is important in choosing the line threshold that the 2 arcs are visible as these correspond to the circles in the original image. "Thresh C" should show the 2nd order radon transform found. "ThresholdC" under the AccArray menu should then threshold the 2nd order Radon transform in the appropriate window. Adjusting the threshold levels will show the maxima in the 2nd order Radon transform. These should correspond to circles in the original image (Leavers, 1992b).

Once the Hough transform for lines and circles has been run (Under the shape menu), the identification of these lines and circles is performed by searching for unique maxima in the constructed arrays. For lines: Select C/L points to about 15. This tells the computer to find 15 local maxima. Now press the "Points" button. This finds those maxima. You can scroll through the found maxima using the "Line Max Find" horizontal scroll bar. The maxima and their θ and p values are displayed. The "Redraw Lin" button will redraw any line selected on from the "Line Max Find " scroll bar. Similarly, circle can be found by selecting say 10 circles on the C/L points scroll bar and then pressing the "Cpoints" button. Circle centers may be scrolled through from the circle centres scroll bar. Please note that if these buttons are pressed before the construction of the matrix and the finding of maxima then the program will display the "Subscript out of range" error message.

Finding unique lines and circles involves searching through the maxima found and comparing different maxima to establish uniqueness. Press "find parm" under the "bact" menu. A message as to the number of unique parallel lines that have been found in the image will appear. These may be scrolled through with the "NoParallel" scroll bar and redraw using the "Redraw Pair" button. Similarly circle may be found by choosing "FindCirc" under the "bact" menu. These can then be scrolled through using the scroll bar next to the "Uniq Circ" button. The redrawing of circles has not been included in the program version on the disc.

Please note that clicking the mouse anywhere on the bacterial co-ordinate box will show the co-ordinates of the mouse at that point. This appears in the top co-ordinate text box.

J.1.2 Program Code

The Program code is subdivided into subroutines, each of which begins with a Sub .... Statement and ends with an End Sub statement. The subroutines names are related to objects on the screen and begin with a characteristic 3 letters:

cmd.. - These refer to buttons on the screen

mnu.. - These refer to menu commands

pct.. - These refer to picture boxes on the screen

txt.. - These refer to the text blocks in which messages appear

hsb.. - Refer to horizontal scroll bars on the screen

```
Option Explicit

Dim histarray1 (0 To 500)

Dim r, g, b, flagpoints As Integer
Dim accumarrayre () As Integer
Dim accumarrayr() As Integer
Dim accumarray() As Integer
Dim accumarrayc() As Integer
Dim max() As Integer
Dim xmax() As Integer
Dim ymax() As Integer
Dim maxc() As Integer
Dim xmaxc() As Integer
Dim ymaxc() As Integer
Dim maxrc() As Integer
Dim xmaxrc() As Integer
Dim ymaxrc() As Integer
Dim xmaxl() As Integer
Dim ymaxl() As Integer
Dim flag() As Integer
Dim xmaxlc() As Integer
Dim ymaxlc() As Integer
Dim zmaxc() As Integer
Dim bactarray()


Sub pcttry_MouseMove (Button As Integer, Shift As Integer, x As Single, y As Single)
End Sub

Sub cmdAuto_Click ()
```

```
If cmdauto.Caption = "AutoOn" Then

        pctaccum.AutoRedraw = True
        cmdauto.Caption = "AutoOff"
ElseIf cmdauto.Caption = "AutoOff" Then
        pctaccum.AutoRedraw = False
        cmdauto.Caption = "AutoOn"
End If
EndSub


Sub cmdcircpoint_Click ()
        Dim x, y, z  As Integer
        ReDim maxc(hsbpoints.Value - 1)
        ReDim xmaxc(hsbpoints.Value - 1)
        ReDim ymaxc(hsbpoints.Value - 1)
        ReDim zmaxc(hsbpoints.Value - 1)
        Dim min, min2, count, counter As Integer
        min = 5
        For count = 0 To hsbpoints - 1
                maxc(count) = 1
        Next count


        For x = 0 To Ubound(accurarrayc, 1) Step 1
        For y = 0 To Ubound(accurarrayc, 2) Step 1
        For z = 0 To Ubound(accurarrayc, 3) Step 1


        If accumarrayc(x, y, z) > min2 Then
        min2 = accumarrayc(x, y, z)

                For count = 0 To hsbpoints - 1
                        If maxc(count) < min2 Then
                                min2 = maxc(count)
                                counter = count
                        End If
                Next count

                maxc(counter) = accumarrayc(x, y, z)
                xmaxc(counter) = x
                ymaxc(counter) = y
                zmaxc(counter) = z
                End If
        Next z
        Next y
        Next x

End Sub
```

```
Sub cmdCThresh_Click ()
Dim count, x, y, counter As Integer
flagpoints = 0
        count = 0

   For x = 0 To Ubound(accumarray, 1) Step 1
   For y = 0 To Ubound(accumarray, 2) Step 1

   If accumarray(x, y) >= hsbnumthresh And accumarray(x, y) < hsens Then
        count = count + 1
   End If
   Next y
   Next x
        ReDim                    max(count                    -                    1)
        ReDim xmax(count - 1)
        ReDim ymax(count - 1)
        counter = 0
   For x = 0 To UBound(accumarray, 1) Step 1
   For y = 0 To UBound(accumarray, 2) Step 1
        If accumarray(x, y) >= hsbnumthresh And accumarray(x, y) < hsens Then
        max(counter) = accumarray(x, y)
        xmax(counter) =  x
        ymax(counter) = y
        counter = counter + 1
        End If
   Next y
   Next x
End Sub

Sub cmdDrawCirc_Click ()
Dim x, y, r As Integer
x = xmaxc (hsbpoints3.Value)
y = ymaxc (hsbpoints3.Value)
r = zmaxc (hsbpoints3.Value)
txtcol.Text = " x = " + Str$ (x) + " y = " + Str$ (y) + " r = " +  Str$(r)
 pctbt.Circle(x, y) , r, RGB(255, 0, 0)
End Sub

Sub cmdLineThresh_Click ()

   Dim x, y As Integer
                pctaccum.ForeColor = RGB (0, 0, 0)

        For x = 0 To UBound(accumarray, 1)
        For y = 0 To UBound(accumarray, 2)
                If accumarray(x, y) >= hsbnumthresh.Value And accumarray(x, y) < hsens.Value Then
                   pctaccum.ForeColor = RGB (255, 255, 255)
                Else
                   pctaccum.ForeColor = RGB(0, 0, 0)
                End If
```

```
        pctaccum.PSet (x, y)

        Next y
        Next x
End Sub

Sub cmdLinMax_Click ()
Dim x, y, count
For y = 1 To UBound(accumarray, 2) - 1
For x = 1 To UBound(accumarray, 1) - 1
If accumarray(x, y) > 10 And accumarray(x, y) > accumarray (x - 1, y) And accumarray(x, y) > accumarray(x + 1, y) And
accumarray(x, y) > accumarray(x, y - 1) And accumarray(x, y) > accumarray(x, y + 1) Then
count = count + 1
End If
Next x
Next Y
txtcol.Text = "Max no: " + Str$(count)
End Sub


Sub cmdlinRedraw_Click ()
Dim x, y, theta, p As Integer
theta = xmax(hsbpoints2.Value)
p = ymax(hsbpoints2.Value)
If theta < 10 Or theta> 170 Then
        For y = -pctbt.ScaleHeight / 2 To pctbt.ScaleHeight / 2
        x = - (p - y * Sin (theta * 3.1415927 / 180)) / (Cos(theta * 3.1415927 / 180))
        If x > - pctbt.ScaleWidth / 2 and x < pctbt.Scalewidth/2 Then
        pctbt.PSet (pctbt.ScaleWidth /2 + x, pctbt.ScaleHeight / 2 + y), RGB (0, 255, 0)
                End If
   Next y
Else
        For x = - pctbt.ScaleWidth / 2 To pctbt.ScaleWidth / 2
            y = - (p - x * Cos(theta * 3.1415927 /180)) / (Sin(theta * 3.1415927 / 180))
            If y > - pctbt.ScaleHeight / 2 And y < pctbt.ScaleHeight / 2 Then
            pctbt.PSet (pctbt.ScaleWidth / 2 + x, pctbt.ScaleHeight / 2 + y), RGB (0, 255, 0)
            End If
        Next x
End If
End Sub

Sub cmdParRedraw_Click()
Dim x, y, theta, p As Integer
theta = ymaxl (hsbparlin.Value, 0)
p = ymaxl (hsbparlin.Value, 1)
If theta < 10 Or theta > 170 Then
        For y = -pctbt.ScaleHeight / 2 To pctbt.ScaleHeight / 2
            x =  - (p - y * Sin(theta * 3.1415927 / 180)) / (Cos(theta * 3.1415927 / 180))
            If x > - pctbt.ScaleWidth / 2 And x < pctbt.ScaleWidth / 2 Then
            pctbt.PSet (pctbt.ScaleWidth / 2 + x, pctbt.ScaleHeight / 2 + y), RGB (0, 255, 0)
```

```
            End If
        Next y
Else
        For x = -pctbt.ScaleWidth / 2 To pctbt.ScaleWidth / 2
            y = - (p - x * Cos(theta * 3.1415927 / 180)) / (Sin(theta * 3.1415927 / 180))
            If y > -pctbt.ScaleHeight / 2 And y < pctbt.ScaleHeigth / 2 Then
         pctbt.PSet (pctbt.ScaleWidth / 2 + x, pctbt.ScaleHeight / 2 + y), RGB (0, 255, 0)
            End If
        Next x
End If

Theta = ymaxl(hsbparlin.value,3)
        P = ymaxl(hsbparlin.Value, 4)
        If theta < 10 or Theta > 170 then
                For y = - pctbt.Scaleheight / 2 to pctbt.Scaleheight /2
                x = - (p-y  * Sin(theta * 3.1415927 / 180))/(Cos(Theta*3.1415927/180))
                If x > - pctbt.Scalewidth / 2 and x < pctbt.Scalewidth/2 then
                Pctbt.Pset (pctbt.scalewidth/2 + x, pctbt.Scaleheight/2 + y), RGB (0,255,0)
                End if
                Next y
        Else
                For x = -pctbt.Scalewidth / 2 To pctbt.Scalewidth/2
                Y = - (p – x * Cos(theta*3.1415927/180))/(Sin(theta*3.1415927/180))
                If y > - pctbt.ScaleHeight / 2 and y < pctbt.Scaleheight / 2 then
                Pctbt.Pset ( pctbt.Scalewidth/2 + x, pctbt.Scaleheight/2 + y) , RGB (0,255,0)
                End If
                Next x
End If
End Sub

Sub cmdPoints_Click ()
        flagpoints = 1
        Dim x, y As Integer
        ReDim max (hsbpoints.Value - 1)
        ReDim xmax (hsbpoints.Value - 1)
        ReDim ymax (hsbpoints.Value - 1)
        Dim min, min2, count, counter As Integer
        min = 5
        For count = 0 To hsbpoints - 1
                max (count) = 1
        Next count

        For x = 0 To UBound (accummarray, 1) Step 1
        For y = 0 To UBound (accummarray, 2) Step 1
        If accumarray (x, y) > min2 Then
        min2 = accumarray (x, y)
        For count = 0 To hsbpoints - 1
                If max (count )< min2 Then
                   min2 = max (count)
                   counter = count
```

```
                End If
        Next count
        max (counter) = accumarray (x, y)
        xmax (counter) = x
        ymax (counter) = y
End If
Next y
Next x
        For count = 0 To UBound (ymax, 1)
                If ymax (count) > pctbt.ScaleWidth Then
                   ymax (count) = - (-ymax(count) + (2 * pctbt.ScaleWidth))
                End If
        Next count
End Sub

Sub cmdRadCircPoints_Click ()
        Dim x, y As Integer
        ReDim maxrc (hsbradcircpoint.Value - 1)
        ReDim xmaxrc (hsbradcircpoint.Value - 1)
        ReDim ymaxrc (hsbradcircpoint.Value - 1)
        Dim min, min2, count, counter As Integer
        min = 5
        For count = 0 To hsbradcircpoint.Value - 1
           maxrc (count) = 1
        Next count
        For x = 0 To UBound (accumarrayr, 1) Step 1
        For y = 0 To UBound  (accumarrayr, 2) Step 1
        If accumarrayr (x, y) > min2 Then
           min2 = accumarrayr (x, y)
           For count = 0 To hsbradcircpoint - 1
                If maxrc (count) < min2 Then
                   min2 = maxrc (count)
                   counter = count
                End If
           Next count
           maxrc (counter) = accumarrayr (x, y)
           xmaxrc (counter) = x
           ymaxrc (counter) = y
        End If
        Next Y
        Next x
           For count = 0 To UBound (ymaxrc, 1)
                If ymaxrc (count) > pctredraw.ScaleWidth Then
                   ymaxrc (count) = -(-ymaxrc (count) + (2 * pctredraw.ScaleWidth))
                End If
        Next count
End Sub

Sub cmdRedraw_Click ()
If flagpoints = 1 Then
```

```
Dim x, y, count, flag As Integer
Dim m1, m As Long
Dim theta, r
pctredraw.ForeColor = RGB (255, 255, 255)
For count = 0 To UBound (max, 1)
    theta =xmax (count)
    r = ymax (count) + pctbt.ScaleWidth
    If theta > 30 And theta < 150 Then
        For x = 0 To pctredraw.ScaleWidth * 5
            y = (r - x * Cos(theta * 3.141593 / 180)) / (Sin(theta * 3.141593 / 180))
            pctredraw.PSet ((x + pctredraw.ScaleWidth / 2) / 5, (y + pctredraw.ScaleHeight) / 5), RGB (255, 255, 255)
        Next x
    Else
        For y = 0 To pctredraw.ScaleHeight * 5
            x = (r - y * Sin(theta * 3.141593 / 180)) / (Cos(theta * 3.141593 / 180))
            pctredraw.PSet ((x + pctredraw.ScaleWidth / 2) / 5, (y + pctredraw.ScaleHeight) / 5), RGB (255, 255, 255)
        Next y
    End If
Next count
Else
txtcol.Text = "Click points first!"
End If
End Sub

Sub cmdUniqCirc_Click ()
Dim x, y, r As Integer
x = ymaxlc (hsbuniqcirc.Value, 0)
y = ymaxlc (hsbuniqcirc.Value, 1)
r = ymaxlc ( hsbuniqcirc.Value, 2)
pctbt.Circle (x, y), r, RGB (255, 0, 0)
End Sub

Sub Form_Load ()
flagpoints = 0
End Sub

Sub hsbhislin_Change ()
txthist.Text = Str$ (hsbhistlin.Value) + ":" + Str$ (histarrayl (hsbhistlin.Value))
End Sub

Sub hsbNumThresh_Change ()
        If hsbnumthresh.Value >= hsens.Value Then hsens.Value = hsbnumthresh.Value + 1
        txtcol.Text = "Low =" + Str$ (hsbnumthresh.Value) + " High = " + Str$ (hsens.Value)
End Sub

Sub hsbParLin_Change ()
If hsbparlin.Value <= UBound (ymax1, 1) Then
txtcol.Text = "Pair no " + Str$ (hsbparlin.Value) + " at x,y,l;x,y,l: " + Str$ (ymaxl (hsbparlin.Value, 0)) + Str$ (ymaxl
(hsbparlin.Value, 1)) + Str$ (ymaxl (hsbparlin.Value, 2 )) + ";"+ Str$ (ymaxl (hsbparlin.Value, 3 )) + Str$ (ymaxl (hsbparlin.Value,
4)) + Str$ (ymaxl (hsbparlin.Value, 5))
```

```
Else
txtcol.Text = "Maximum number pairs reached"
hsbparlin.Value = UBound (ymaxl,1)
End If
End Sub

Sub hsbpoints_Change ()
        txtcol.Text = "Number of points to find = " + Str$ (hsbpoints.Value)
End Sub

Sub hsbpoints2_Change ()
If hsbpoints2.Value <= UBound (max, 1)
        txtcol.Text = "max (" + Str$(hsbpoints2.Value) + ") = " + Str$ (max(hsbpoints2.Value)) + " at " + Str$
(xmax(hsbpoints2.Value)) + ", " + Str$ (ymax(hsbpoints2.Value))
Else
        hsbpoints2.Value = UBound (max, 1)
End If
End Sub

Sub hsbpoints3_Change ()
If hsbpoints3.Value <= UBound (maxc, 1) Then
        txtcol.Text = "max (" + Str$(hsbpoints3.Value) + ") = " + Str$ (maxc(hsbpoints3.Value)) + " at " +
Str$(xmaxc(hsbpoints3.Value)) + " , " + Str$(ymaxc(hsbpoints3.Value)) + ", radius = " + Str$(zmaxc(hsbpoints3.Value))
Else
        hsbpoints3.Value = UBound (maxc, 1)
        txtcol.Text = "upper limit"
End If
End Sub

Sub hsbRadCircPoint_Change ()
txtcol.Text = "Number of points to find = " + Str$ (hsbradcircpoint.Value)
End Sub

Sub hsbrcmaxfind_Change ()
If hsbrcmaxfind.Value <= Ubound(maxrc,1) Then
txtcol.Text = "max(" + Str$(hsbrcmaxfind.Value) + ") = " + Str$(maxrc(hsbrcmaxfind.Value)) + " at " +
Str$(xmaxrc(hsbrcmaxfind.Value)) + " , " + Str$(ymaxrc(hsbrcmaxfind.Value))
Else
        hsbrcmaxfind.Value = UBound (maxrc, 1)
End If
End Sub

Sub hsbUniqCirc_Change ()
If hsbuniqcirc.Value <= UBound(ymaxlc, 1) Then
txtcol.Text = "Circle center no " + Str$(hsbuniqcirc.Value) + " at x,y,r: " + Str$(ymaxlc(hsbuniqcirc.Value, 0))
+str$(ymaxlc(hsbuniqcirc.Value, 1)) + Str$(ymaxlc(hsbuniqcirc.Value, 2))
Else
txtcol.Text = "Maximum number circles reached"
hsbuniqcirc.Value = UBound (ymaxlc, 1)
End If
```

```
End Sub

Sub hsens_Change ()
        If hsbnumthresh.Value >= hsens.Value Then hsens.Value = hsbnumthresh.Value + 1
        txtcol.Text = "Low =" + Str$(hsbnumthresh.Value) + " High = " + Str$(hsens.Value)
End Sub

Sub mnuaccum0_click ()
Dim x,y As Integer
For x = 0 To 180
For y = 0 To 700
accumarray(x, y) = 0
Next y
Next x
End Sub

Sub mnuaccumC_Click ()
mnuaccum0_click
Dim count, theta, p, plot
for count = 0 To UBound(xmax, 1)
                For theta = 0 To 180 Step 1
                p = ((xmax(count) - 350) * Cos(theta * 3.14159265 / 180)) + (( ymax(count) - 350) * Sin(theta * 3.14159265
/ 180))
                plot = p + 350
                If 0 < plot And plot < 700 Then
                accumarray(theta, plot) = accumarray(theta, plot) + 1
                End If
                Next theta
Next count
End Sub

Sub mnuBactFind_Click ()
Dim n, count, counter2, counter, thetac, pc, x1, x2, y1, y2, m1, c, theta, theta2, p, p2 As Integer
ReDim bactarray (0 To UBound(ymaxl, 1), 0 To 2)
n = UBound(ymaxlc, 1)
ReDim flag (0 To n)
For count = 0 To n
flag (count) = 1
Next count
For count = 0 To UBound(ymaxl, 1)
bactarray(count, 0) = count
theta = ymaxl(count, 0)
theta2 = ymaxl(count, 3)
p = ymaxl (count, 1)
p2 = ymaxl (count, 4)
For counter = 0 To n - 1
For counter2 = counter + 1 To n
x1 = ymaxlc(counter, 0) - pctbt.ScaleWidth / 2
y1 = pctbt.ScaleHeight / 2 - ymaxlc(counter, 1)
x2 = ymaxlc(counter2, 0) - pctbt.ScaleWidth / 2
```

```
y2 = pctbt.ScaleHeight / 2 - ymaxlc(counter2, 1)
m1 = (x2 - x1) / y1 - y2)
c = y1 - m1 * x1
thetac = (180 / 3.1415927) * Atn(m1)
pc = (c / (1 +m1) ^ 2) * Sqr((1 + m1 ^ 2))
        If Abs (thetac - theta) < 6 Then
            flag (counter) = 0
            flag (counter2) = 0
            bactarray (count, 1) = counter
            bactarray (count, 2) = counter2
        End If
Next counter2
Next counter
Next count
txtcol.Text = "bact:" + Str$(bactarray(0, 0)) + Str$(bactarray(0, 1)) + Str$(bactarray(0, 2))
End Sub

Sub mnuCopy_Click ()
clipboard.Clear
clipboard.SetData pctredraw.Image
End Sub

Sub mnuellipsthresh_Click ()
   Dim x, y As Integer
                pctellipse.ForeColor = RGB(0, 0, 0)
        For x = 0 To UBound(accumarrayre, 1)
        For y = 0 To UBound(accumarrayre, 2)
                If accumarrayre(x, y) >= hsbnumthresh.Value And accumarrayre(x, y) < hsens.Value Then
                    pctellipse.ForeColor = RGB(255, 255, 255)
                Else
                    pctellipse.ForeColor = RGB(0, 0, 0)
                End If
        pctellipse.PSet (x, y)
        Next y
        Next x
End Sub

Sub mnuExit_Click ()
        End
End Sub

Sub mnuFindCirc_Click ()
        Dim x, y As Integer
        Dim count, counter, i, j, k As Integer
        ReDim flag(0 To UBound(xmaxc, 1) + 1) As Integer
        ReDim xmaxlc(0 To UBound(xmaxc, 1), 0 to 2)
        For i = 0 To UBound(xmaxc, 1)
            flag(i) = 1
        Next i
        j = 1
```

```
        count = 1
        counter = 0
        For k = 0 To UBound(xmaxc, 1)
        For i = k + 1 To UBound(xmaxc, 1)
        If Abs (ymaxc(i) - ymaxc(k)) <= 3 And Abs (xmaxc(i) - xmaxc(k)) <= 3 Then
            flag(i) = 0
        End If
        Next I
        Next K
        For I = 0 To UBound(xmaxc, 1)
        If flag(i) = 1 Then
                xmaxlc(i, 0) = xmaxc(i)
                xmaxlc(i, 1) = ymaxc(i)
                xmaxlc(i, 2) = zmaxc(i)
                count = count + 1
                flag(i) = 0
        End If
        Next I
        If count = 1 Then
        txtcol.Text = "One unique circle."
        Else
        ReDim ymaxlc(0 To count - 1, 0 To 2)
        For k = 0 to UBound(xmaxlc, 1)
        If xmaxlc(k, 1) > 0 Then
        ymaxlc(counter, 0) = xmaxlc(k, 0)
        ymaxlc(counter, 1) = xmaxlc(k, 1)
        ymaxlc(counter, 2) = xmaxlc(k, 2)
        counter = counter + 1
        End If
        Next k
        txtcol.Text = Str$(UBound(ymaxlc, 1)) + " Unique circles found."
        End If
End Sub

Sub mnuHelpLin_Click ()
MsgBox "To find lines click lines under shape menu.  Then check number of lines required on the C/L points bar.  Scroll through these values on the line max find bar", 0, "Help"
End Sub

Sub mnuHlinHist_Click ()
Dim x, y, count As Integer
For count = 0 To 500
histarrayl(count) = 0
Next count

For count = 0 To 500
For x = 0 To UBound(accumarray, 1)
For y = 0 To UBound(accumarray, 2)
If accumarray(x, y) = count Then histarrayl(count) = histarrayl(count) + 1
Next y
```

```
Next x
Next count
End Sub

Sub mnuhoughCirc_Click ()
screen.MousePointer = 11
Dim x, y, a, b, d, theta
ReDim accumarrayc(pctbt.ScaleWidth, pctbt.ScaleHeight, 20)
txtcol.Text = "Working .... "
For x = 0 To pctbt.ScaleWidth
For y = x To pctbt.ScaleHeight
If pctbt.Point(x, y) > 0 Then
        For theta = 91 To 270
        For a = 0 To pctbt.ScaleWidth
            b = Tan(theta * 3.1415927 / 180) * a + ( y - x * Tan(theta * 3.1415927 / 180))
            If b < pctbt.ScaleHeight And b > 0 And Sqr((b - y) ^ 2 + (a - x) ^ 2) < 20 And Sqr((b - y) ^ 2 + (a - x) ^ 2) > 3 Then
            d = Sqr((b - y) ^ 2 + (a - x) ^ 2)
            accumarrayc(a, b, d) = accumarrayc(a, b, d) + 1)
            End If
        Next a
        Next theta
Next y
Next x
txtcol.Text = " Finished"
screen.MousePointer = 1
End Sub

Sub mnuHoughlin_Click ()
screen.MousePointer = 11
Dim a, b, p, plot, x, y, x0, y0, xchange, ychange, direc, theta, sens, I As Integer
Dim pow, col, remain As Long
Dim count, count2, average As Integer
ReDim accumarray(0 To 180, 2 * pctbt.ScaleWidth) As Integer
For a = 0 To 180
For b = 0 To pctbt.ScaleWidth
    accumarray(a, b) = 0
Next b
Next a
txtcol.Text = "Working..... "
For x = 0 To pctbt.ScaleWidth Step 1
For y = 0 To pctbt.ScaleHeight Step 1
        If pctbt.Point(x, y) > 0 Then
            ychange = pctbt.ScaleHeight / 2 - y
            xchange = x - pctbt.ScaleWidth / 2
            For theta = 0 To 180 Step 1
                p = (xchange * Cos(3.14159265 * theta / 180)) + (ychange * Sin(3.14159265 * theta / 180))
                If p < UBound(accumarray, 2) And p > 0 Then
                accumarray(theta, p) = accumarray(theta, p ) + 1
            ElseIf p > -UBound(accumarray, 2) And p < 0 Then
                accumarray(theta, 2 * pctbt.ScaleWidth + p) = accumarray(theta, 2 *pctbt.ScaleWidth + p) + 1
```

```
            End If
        Next theta
    End If
Next y
Next x
txtcol.Text = "Finished"
screen.MousePointer = 1
End Sub

Sub mnuparm_Click ()
        Dim x, y As Integer
        Dim count, counter, I, j, k As Integer
        ReDim flag(0 To UBound(xmax, 1) + 1) As Integer
        ReDim xmaxl(0 To UBound(xmax, 1), 0 to 5)
        For I = 0 to UBound(xmax, 1)
            flag(i) = 1
        Next I
        xmaxl (0, 0) = xmax(0)
        xmaxl (0, 1) = ymax(0)
        xmaxl (0, 2) = max(0)
        j = 1
        flag(0) = 0
        count = 0
        counter = 0
        For k = 1 To UBound(xmax, 1)
        For I = k + 1 To UBound(xmax, 1)
        If Abs (ymax(i) - ymax(k)) < 7 And Abs(xmax(i) - xmax(k)) < 3 Or Abs(xmax(i) - xmax(k)) > 175) Then
         flag(i) = 0
        End If
        Next I
        Next K
        Do
        For I = j To UBound(xmax, 1)
        If flag(i) = 1 Then
            If Abs(xmax(i) - xmax(j - 1)) < 3 And Abs(max(i) - max(j - 1)) < 20 And Abs(ymax(i) - ymax(j - 1)) > 3 And
Abs(ymax(i) - ymax(j - 1)) < 25 Then
                    xmaxl(j - 1, 3) = xmax(i)
                    xmaxl(j - 1, 4) = ymax(i)
                    xmaxl(j - 1, 5) = max(i)
                    count = count + 1
                    flag(i) = 0
                    Exit For
            End If
         End If
        Next I
        For k = 1 To Ubound(xmax, 1) + 1
            j = k
            If flag(j) = 1 Then
            xmaxl(j, 0) = xmax(j)
            xmaxl(j, 1) = ymax(j)
```

```
            xmaxl(j, 2) = max(j)
            flag(j) = 0
            j = j + 1
            Exit  For
            End If
        Next k
        Loop While j <= UBound(xmax, 1)
        If count = 0 Then
        txtcol.Text = "No pairs paralell lines found."
        Else
        ReDim ymaxl(0 To count - 1, 0 To 5)
        For k = 0 To UBound(xmaxl, 1)
        If xmaxl (k, 2) > 0 And xmaxl(k, 5) > 0 Then
        ymaxl (counter, 0) = xmaxl (k, 0)
        ymaxl (counter, 1) = xmaxl (k, 1)
        ymaxl (counter, 2) = xmaxl (k, 2)
        ymaxl (counter, 3) = xmaxl (k, 3)
        ymaxl (counter, 4) = xmaxl (k, 4)
        ymaxl (counter, 5) = xmaxl (k, 5)
        counter = counter + 1
        End If
        Next k
        txtcol.Text = Str$(count) + " Pairs parallel lines found"
        End If
End Sub

Sub mnuPaste_Click ()
pctbt.Picture = clipboard.GetData ()
End Sub

Sub mnuRadonc_Click ()
screen.MousePointer = 11
Dim a, b, p, plot, x, y, x0, y0, xchange, ychange, direc, theta, sens, I As Integer
Dim pow, col, remain As Long
Dim count, count2, average As Integer
ReDim accumarrayr(0 To UBound(accumarray, 1), 0 To UBound(accumarray, 2)) As Integer
For a = 0 To UBound(accumarrayr, 1)
For b = 0 To UBound(accumarrayr, 2)
    accumarrayr(a, b) = 0
Next b
Next a
txtcol.Text = "Working....."
For x = 0 To UBound(accumarray, 1) Step 1
For y = 0 To UBound(accumarray, 2) Step 1
        If accumarray(x, y) > hsbnumthresh And accumarray(x, y) < hsens Then
            For theta = 0 To 180 Step 1
                p = (x * Cos(theta * 3.14159265 / 180)) + (y * Sin(theta * 3.14159265 / 180))
                If p < UBound(accumarrayr, 2) And p > 0 Then
                    accumarrayr(theta, p) = accumarrayr(theta, p) + 1
                End If
```

```
            Next theta
        End If
Next y
Next x
txtcol.Text = "Finished"
screen.MousePointer = 1
End Sub

Sub mnuRadone_Click ()
screen.MousePointer = 11
Dim a, b, p, plot, x, y, x0, y0, xchange, ychange, direc, theta, sens, I As Integer
Dim pow, col, remain As Long
Dim count, count2, average As Integer
ReDim accumarrayre(0 To UBound(accumarrayr, 1), 0 To UBound(accumarrayr, 2)) As Integer
For a = 0 To UBound(accumarrayre, 1)
For b = 0 To UBound(accumarrayre, 2)
    accumarrayre(a, b) = 0
Next b
Next a
txtcol.Text = "Working....."
For x = 0 To UBound(accumarrayr, 1) Step 1
For y = 0 To UBound(accumarrayr, 2) Step 1
        If accumarrayr(x, y) > hsbnumthresh And accumarrayr(x, y) < hsens Then
            For theta = 0 To 180 Step 1
                p = (x * Cos(theta * 3.14159265 / 180)) + (y * Sin(theta * 3.14159265 /180 ))
                If p < UBound(accumarrayre, 2) And P > 0 Then
                    accumarrayre(theta, p) = accumarrayre(theta, p) + 1
                End If
            Next theta
        End If
Next y
Next x
txtcol.Text = "Finished"
screen.MousePointer = 1
End Sub

Sub mnuRemStripe_Click ()
pctbt.Line (0, pctbt.ScaleHeigth - 1) - (pctbt.ScaleWidth, pctbt.ScaleHeight - 1), RGB(0, 0, 0)
End Sub

Sub mnuSavePic_Click ()
SavePicture pctredraw.Image, "c:\vbprog\result.bmp"
End Sub

Sub mnuThreshc_Click ()
    Dim x, y As Integer
                pctredraw.ForeColor = RGB(0, 0, 0)
        For x = 0 To UBound(accumarrayr, 1)
        For y = 0 To UBound(accumarrayr, 2)
            If accumarrayr(x, y) >= hsbnumthresh.Value And accumarrayr(x, y)< hsens.Value Then
```

```
                pctredraw.ForeColor = RGB(255, 255, 255)
            Else
                pctredraw.ForeColor = RGB(0, 0, 0)
            End If
        pctredraw.PSet (x, y)
        Next y
        Next x
End Sub
Sub pctAccum_MouseDown (Button as Integer, Shift As Integer, x As Single, y As Single)

    Dim col As String
    If pctaccum.Point(x, y) > 0 Then
    col = "white"
    Else col = "black"
    End If
    txtcol.Text = "x= " + Str$(x) + " , y = " + Str$(y) + " , " + col
    End Sub

Sub pctBt_MouseDown (Button As Integer, Shift As Integer, x As Single, y As Single)
    Dim col As String
    If pctbt.Point(x, y) > 0 Then
    col = "white"
    Else col = "black"
    End If
    txtcolor.Text = "x= " + Str$(x - pctbt.ScaleWidth / 2) + " , y = " + Str$(pctbt.ScaleWidth / 2 - y) + " , " + col
End Sub

Sub pctRedraw_MouseDown (Button As Integer, Shift As Integer, x As Single, y As Single)
Dim col As String
        If pctredraw.Point(x, y) > 0 Then
        col = "white"
        Else col = "black"
        End If
        txtcolor.Text = "x = " + Str$(x - pctbt.ScaleWidth / 2) + " , y = " + Str$(pctbt.ScaleWidth / 2 - y) + " , " + col
End Sub
```

J.2 OPTIMAS ALI MACRO OF RADON (HOUGH) TRANSFORM

```
// The following macro implements the Hough Transform for finding a line. This is acheived as per the
// paper " Use of the Radon transform as a method of exctracting shape in two dimensions", V F Leavers,
// Image and vision computing, volume 10, no 2, 1992, pp 99-107

// The macro scans the region of interest and outputs the angle, closest distance to the origin, and number of
// votes for the particular line

//Declare variables, maxi is the vector of high votes, p and p2 are the p-values (ie line minimum distance from
// the origin) , theta and theta1 are angles, where theta is the incremented angle in the loop and theta1 is the
// vector of angles corresponding to the angle of the line perpendicular to the line from the origin

integer maxi;
integer p;
integer p2[10];
integer theta1[10];
real theta;
integer counts;                 //counts are the loop variables for while loops
integer counts2;
integer counts3;
integer counter;
integer count;
integer i;
integer j;
integer k;
byte sbytes;
integer maxn[10];
integer flag[11];               //Flag vector for establishing uniqueness
integer parm[10,6];

// initialise variables

maxi = 3;  // This speeds up processing as only places with more than 3 votes are considered
                // as possible lines.
p2[0..10] = 0;          // Vector notation in ALI - this initialises all elements of vector p2 to 0;
theta1 [0..10] = 0;
maxn[0..10] = 0;
flag[0..11] = 1;
parm[0..10,0..6] = 0;

// Now construct sbytes - where sbytes is a 2 dimensional matrix of grey values (0-255) coressponding directly
// to the Region of interest.
getpixelrect(roi,sbytes);

// Now make myroi - which is a 2x2 matrix containing the upperleft and lower right co-ordinates of the Region
// of interest
myroi = (convertcalibtopixels(roi));
```

```
show(myroi);                                    //Shows ROI

// Define the accumulator array as a 180 x ROI width array
integer accumarray[181,myroi[1,1]-myroi[0,1]];

// Set all elements of the accumulator array to zero
accumarray[0..181,0..(myroi[1,1]-myroi[0,1])] = 0;

// now accumulate non-zero's into the accumulator array:
counts = myroi[1,1] - myroi[0,1];
counts2 = 0;
while(counts>0)
{
 while(counts2<myroi[1,0]-myroi[0,0])
 {
  if(sbytes[convertcalibtopixels(roi[0])+counts,convertcalibtopixels(roi[2])+counts2]>0)
    {
      theta = 0;
      p = 0;
       while (theta < 180)
            {
               theta++;
                p = (cos(3.1416*theta/180)*counts2)+(sin(3.1416*theta/180)*counts);
                 if(p>=0)
                       {
                          if(p < (myroi[1,1]-myroi[0,1]))
                            {
                                 accumarray[theta,pi]++;
                            }
                       }
            }
    }
counts2++;
}
counts2 = 0;
counts--;
}

// now Accumulate the maximum vector to the desired number of maxima - this gives the highest number of
// votes for each line. Note that there is no discrimination here for the same line voted slightly differently for.
// Note also that the dimensions of the maxn vector determine the number of lines to find
counts = 0;
counts2 = 0;
maxi = 3;

while(counts<180)
{
 While (counts2<(myroi[1,1]-myroi[0,1]))
{
          if(accumarray[counts,counts2]>maxi)
```

```
        {
          counts3 = 0;
          maxi = accumarray[counts,counts2];
           while (counts3 < 10)
                {
                  if ( maxn[counts3] < maxi)
                    {
                          maxi = maxn[counts3];
                          counter = counts3;
                     }
                counts3++;
                }
          maxn[counter] = accumarray[counts,counts2];
          theta1[counter] = counts;
           p2[counter] = counts2;
           }

counts2++;
}
counts2 =0;
counts++;
}

// Now we search through the line vectors to determine uniqueness and possible parallelity.
// Here user defined variables are important, i.e. distance apart  and theta similarities which define uniqueness,
// Distance apart which decides whether they should be considered as part of the bacteria, similarities in length
// etc, etc.

// first loop for finding uniqueness of lines:
parm[0,0] = theta1[0];
parm[0,1] = p2[0];
parm[0,2] = maxn[0];
flag[0] = 0;
j=1;
k =1;
counter = 0;
count = 0;

while (k < 10 )
        {
          i = k + 1;
          while (i<10)
            {
                if(abs(p2[i]-p2[k])<7)
                  {
                    if(abs(theta1[i]-theta1[k])<3)
                          {
                            flag[i] =0;
                          }
                     if(abs(theta1[i]-theta1[k])>175)
```

```
                                {
                                  flag[i] =0;
                                }
                            }
                        i++;
                        }
                k++;
            }

// Now we establish parallelity from those objects not flagged to zero (if they have been flagged to zero there is
// another line with a similar angle and p value – i.e. that defines it's uniqueness.

While (j < 10)                  //Check that this should not be 11
{
 i =j;
  while (i < 10)
            {
             if(flag[i]=1)
               {
                 if(abs(theta1[i] - theta1[j-1])<3)
                   {
                      if(abs(maxn[i]-maxn[j-1])<20)
                        {
                         if(abs(p2[i]-p2[j-1])>3)
                           {
                                if(abs(p2[i]-p2[j-1])<25)
                                  {
                                    parm[j-1,3] = theta1[i];
                                     parm[j-1,4] = p2[i];
                                     parm[j-1,5] = maxn[i];
                                      count++;
                                      flag[i] =0;
                                  }
                           }
                        }
                   }
               }
            i++;
    }
k=1;
while(k<10)
  {
    j = k;
            if(flag[j] = 1)
             {
               parm[j,0] = theta1[j];
               parm[j,1] = p2[j];
               parm[j,3] = maxn[j];
               flag[j] = 0;
               j++;
```

```
            k = 10;
            }
    k++;
    }
}

// Now report number parallel lines  - these correspond to bacteria
integer sa = parm[0,] > 0;                    // Selector vector of true/false for pairs parm lines - see pg 417 manual
ntrue = vectorlength(SqueezeSelector(sa));
macromessage("Number paralell lines = ",ntrue);
```

APPENDIX K: LENGTH VARIATION

Variation in length was reported for all trials. The figures below show these lengths.

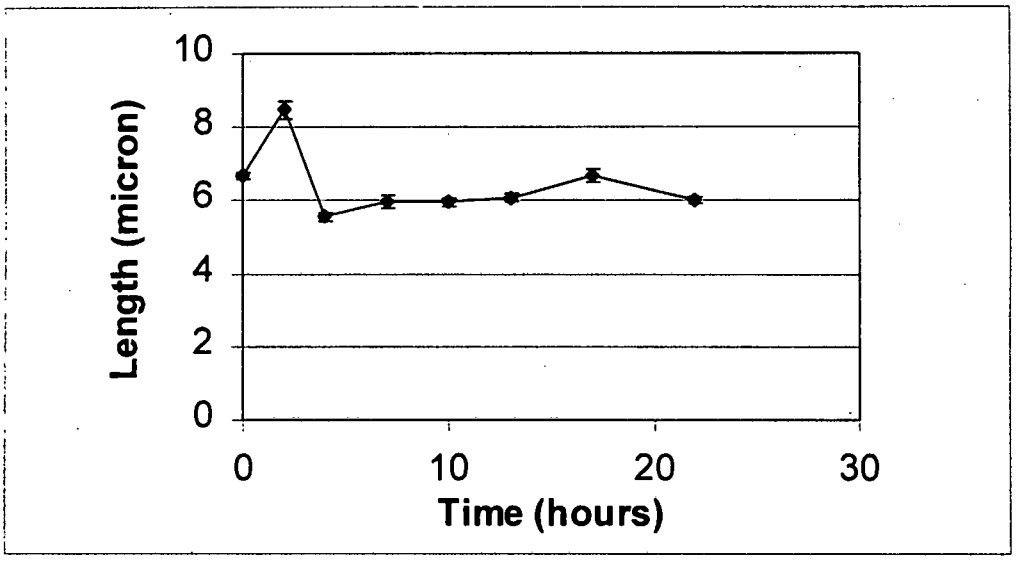


Figure K.1: *Length variation for large shakeflask*

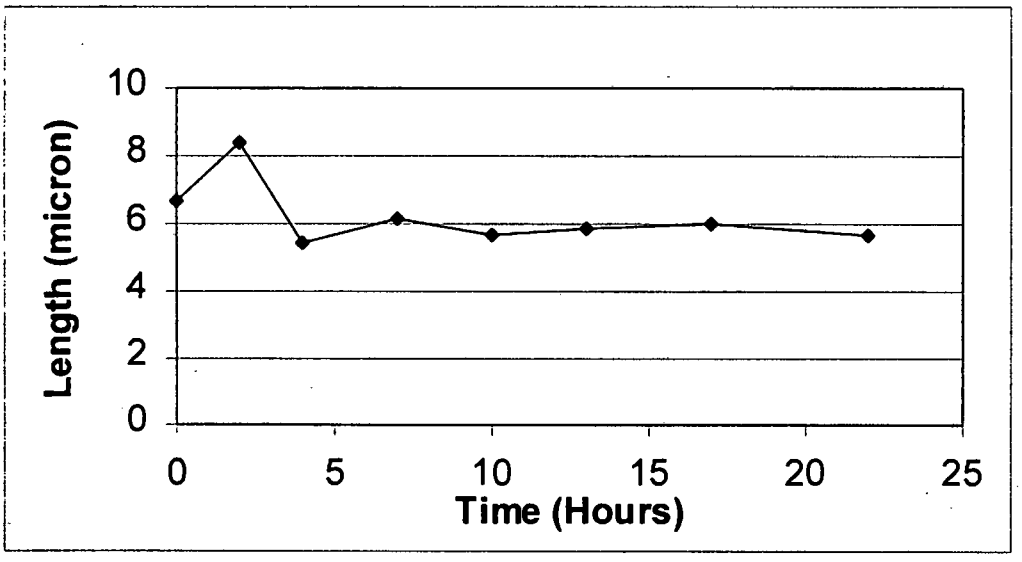


Figure K.2: *Length variation for medium shakeflask*

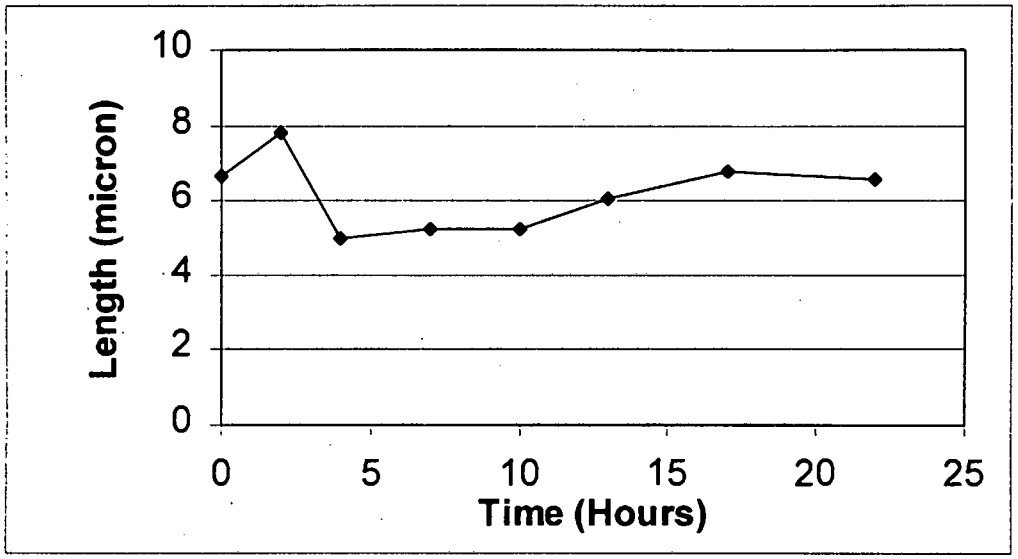


Figure K.3: *Length variation for small shakeflask*

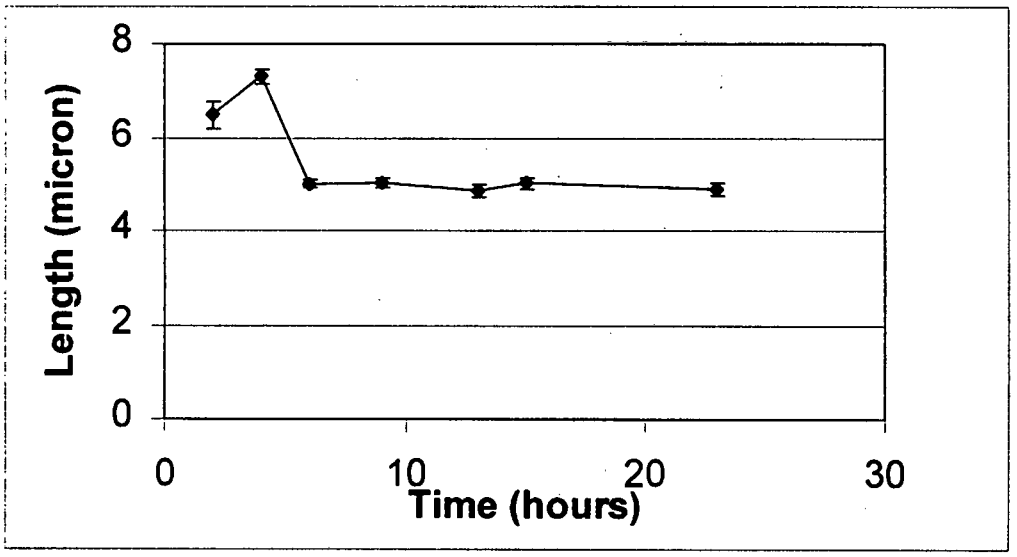


Figure K.4: *Length variation for nutrient media trial.*

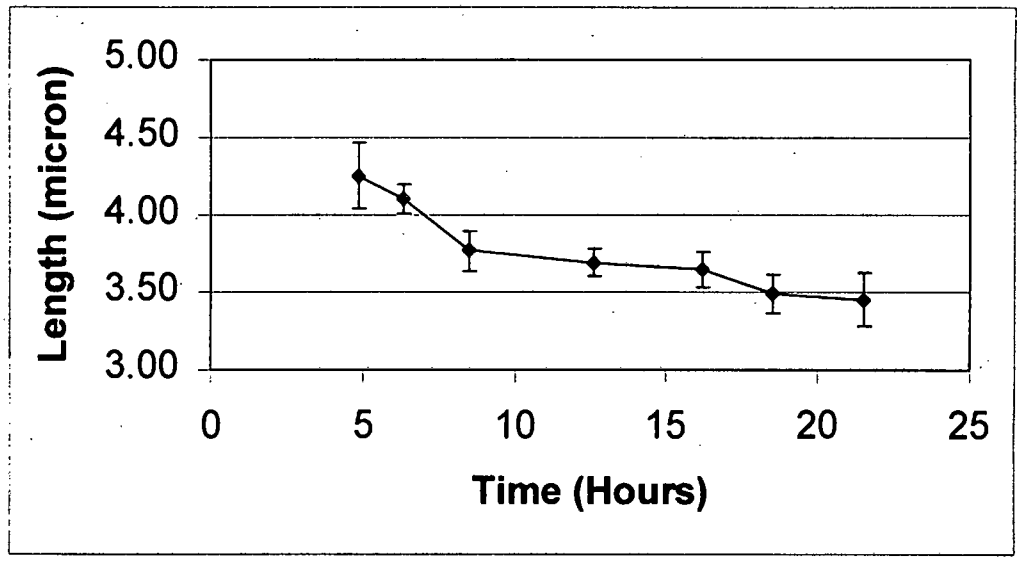


Figure K.5: *Length variation for full fermenter trial*

APPENDIX L: MANUAL VS AUTO COUNTS

Graphs showing the comparison between manual and automatic counts follow. Included on each graph is the equation and R^2 value of a linear fit of automatic counts plotted vs manual counts.

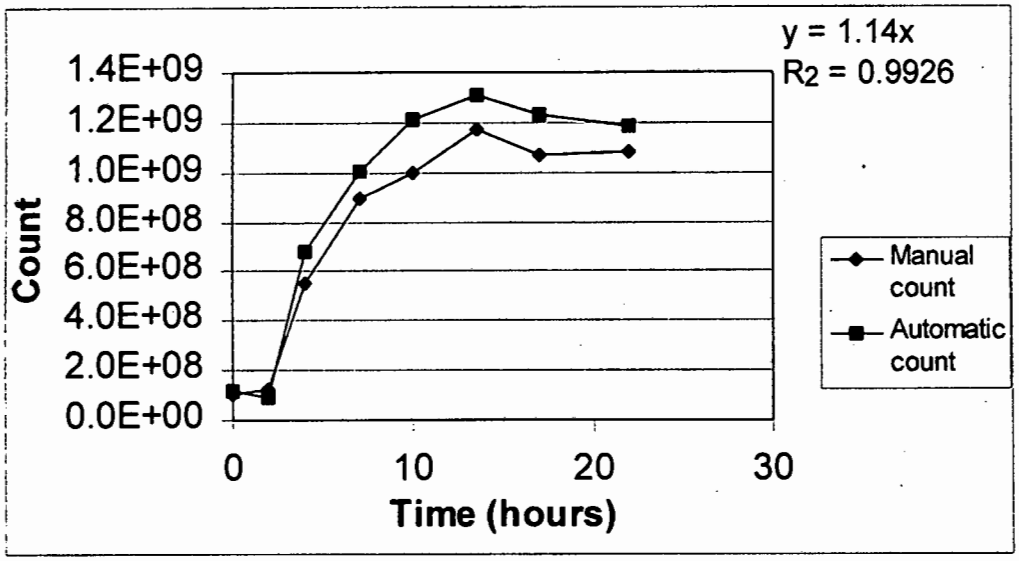


Figure L.1: *Manual vs Automatic count for large shake flask.*

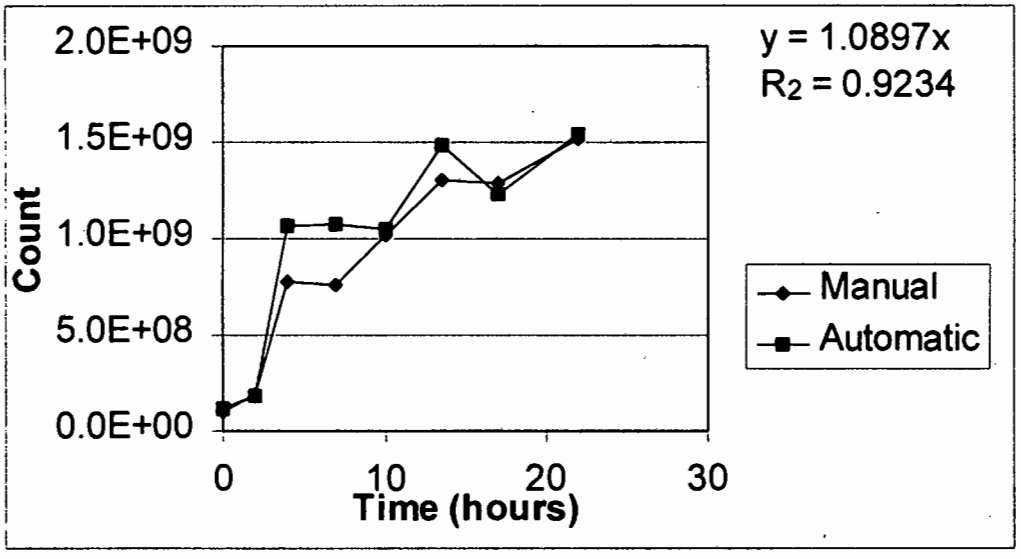


Figure L.2: *Manual vs automatic count for medium shake flask.*

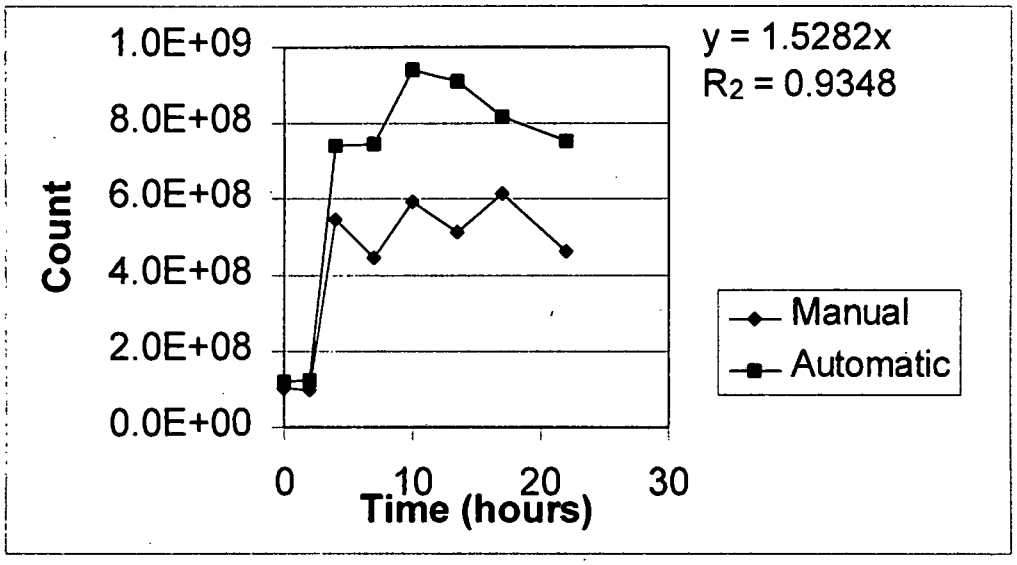


Figure L.3: *Manual vs automatic count for small shake flask.*

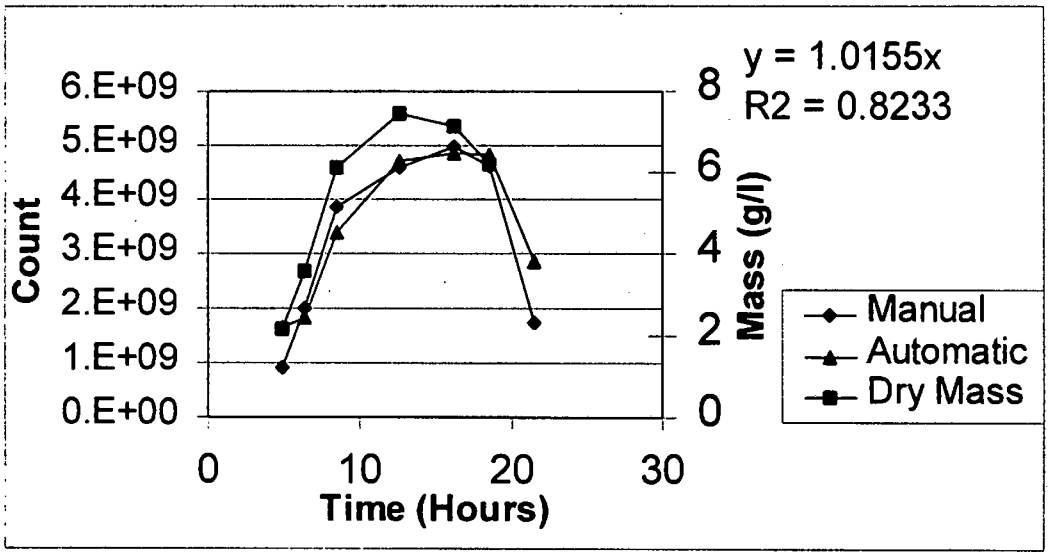


Figure L.4: *Manual vs automatic count for Full Fermentation*

APPENDIX M: MANUAL VALIDATION OF AUTOMATICALLY FOUND AREAS

Table M.1: *Manual validation of automatically found cells*

Hour:	5	7	9	13	16	18
Total cells: (manual)	131	375	201	182	105	99
Cells correct: (including tendrils)	122	339	190	174	100	91
Cells not found	9	36	11	8	5	8
Areas not cells but identified as such	2	5	6	4	5	5
Tendrils influencing breadth	8	21	14	5	9	6
Cells correct: (including tendrils), %	93.12977	90.4	94.52736	95.6044	95.2381	91.91919
Cells not found, %	6.870229	9.6	5.472637	4.395604	4.761905	8.080808
Areas not cells but identified as such, %	1.612903	1.453488	3.061224	2.247191	4.761905	5.208333
Tendrils influencing breadth, %	6.557377	6.19469	7.368421	2.873563	9	6.593407

Table M.2: *Manual validation of the automatic identification of spores and proteins*

Photo Number	Manual Spore	Auto Spore	Spores Not found	Spores incorrect	Manual Prot	Auto Prot	Prot not found	Prot incorrect
0	1	1	0	0	2	2	0	0
1	0	0	0	0	0	0	0	0
2	1	1	0	0	1	1	0	0
3	1	1	0	0	1	0	1	0
4	4	4	0	0	2	1	1	0
5	1	1	0	0	1	0	1	0
6	1	1	0	0	1	1	0	0
7	3	3	0	0	3	3	0	0
8	1	1	0	0	0	0	0	0
9	3	3	0	0	3	2	0	0
10	2	2	1	1	0	1	0	0
11	3	3	0	0	3	3	0	0
12	1	1	0	0	1	1	0	0
13	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0
15	2	2	0	0	2	2	0	0
16	2	2	0	0	2	2	0	0
17	2	2	0	0	2	1	1	0
18	1	0	1	0	0	0	0	0
19	0	0	0	0	0	0	0	0
20	1	1	0	0	2	1	1	0
21	2	2	0	0	3	3	0	0
22	3	3	0	0	3	2	1	0
23	3	3	0	0	2	2	0	0
24	3	2	1	0	3	2	1	0
25	6	7	0	1	6	4	2	0
26	2	1	1	0	2	1	1	0
27	0	0	0	0	0	0	0	0
28	2	2	0	0	3	3	0	0
29	2	2	0	0	2	2	0	0
TOTAL:	**53**	**51**	**4**	**2**	**50**	**40**	**10**	**0**

APPENDIX N: TRENDS IN AREA

Cell area was largely a function of cell length. Cell area are reported below.

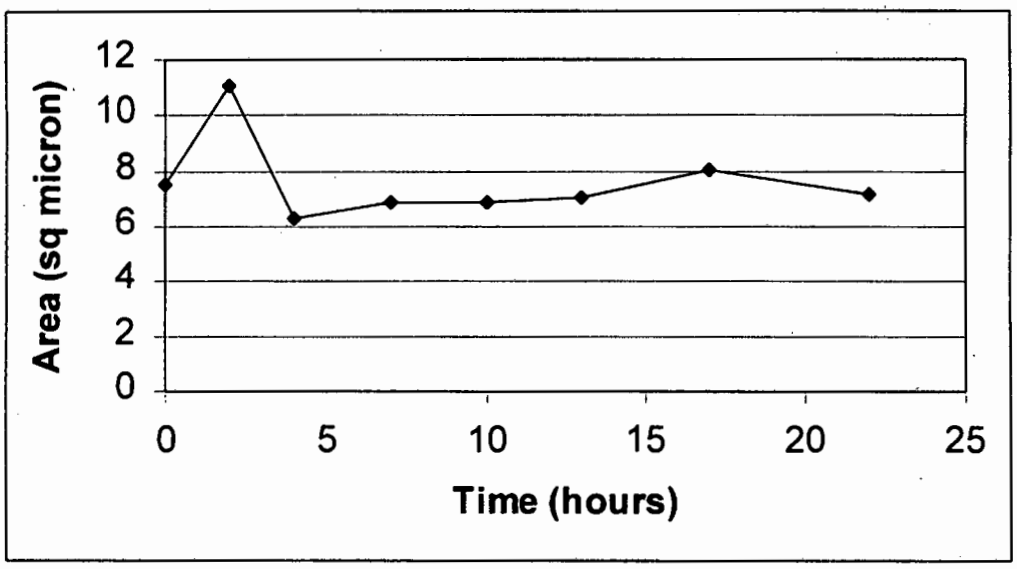


Figure N.1: *Cell area for large shakeflask*

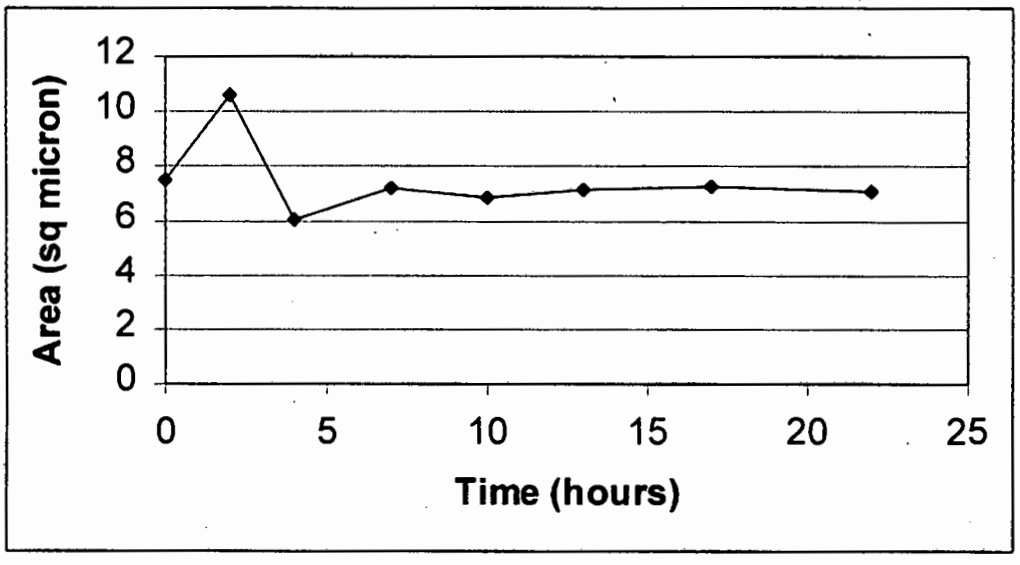


Figure N.2: *Cell area for medium shakeflask*

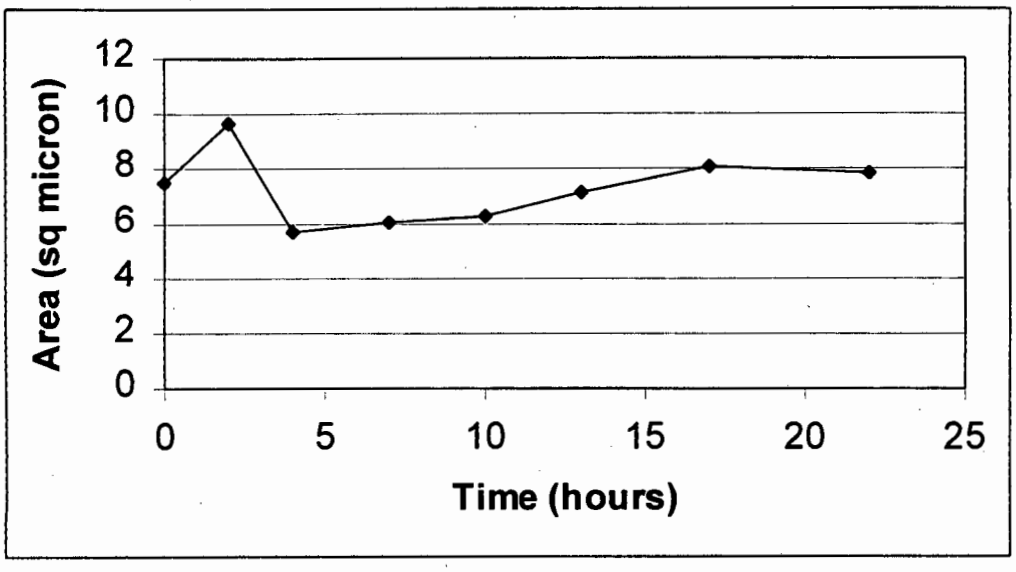


Figure N.3: *Cell area for small shakeflask*

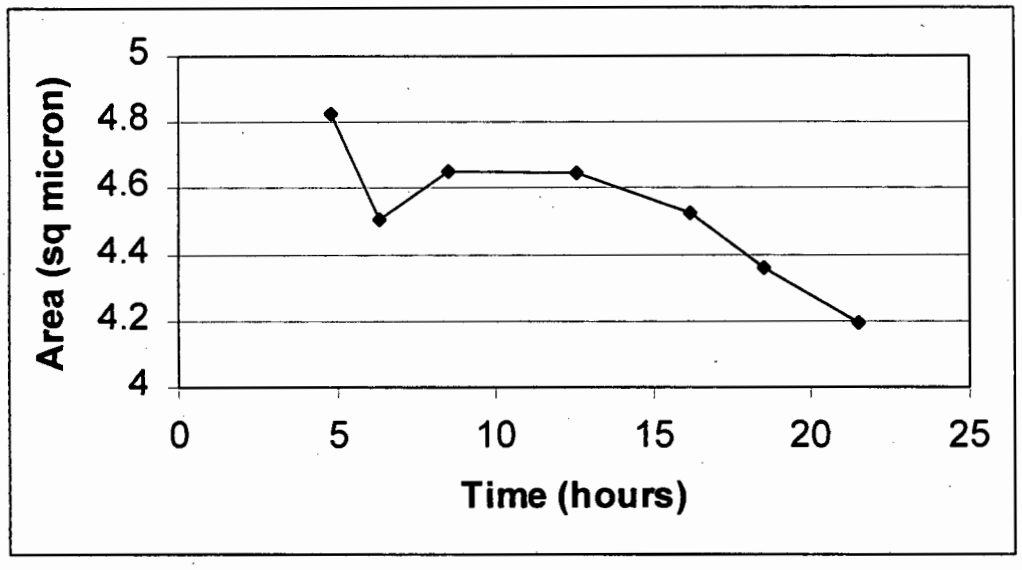


Figure N.4: *Cell Area for full fermentation*

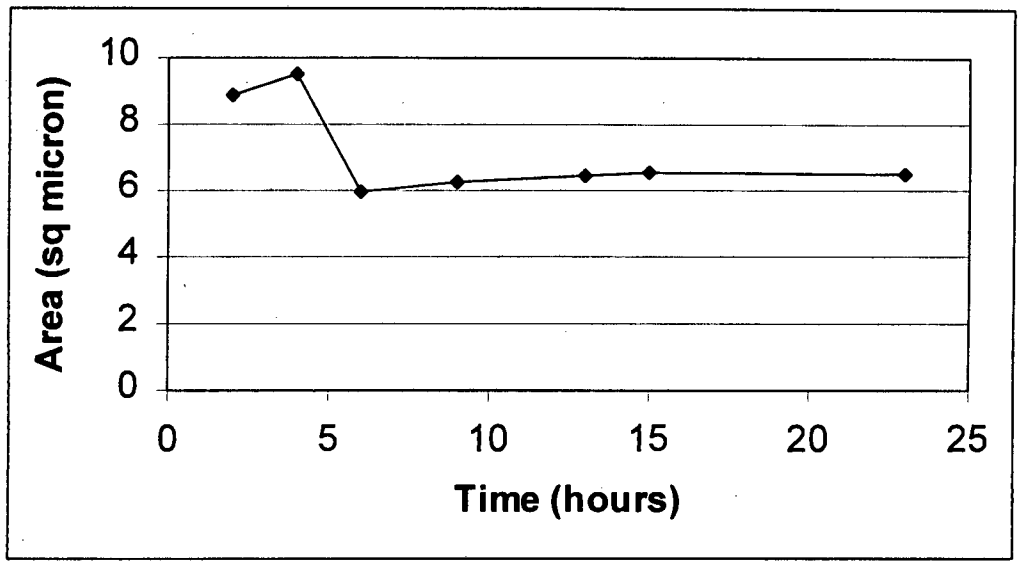


Figure N.5: *Cell area for nutrient broth shake flask trial*

APPENDIX O: MOTION STUDIES

The reported motility and individual bacterial motion is reported below. Table O.1 details the movements of the 8 bacteria followed in the first image sequence in terms of pixel co-ordinates and gives the velocity between frames in microns.

***Table O.1**: Velocity of individual bacteria in sample 1*

Cell No:	Cell 1		Distance	Cell No:	Cell 2		Distance
Photo Num	Pixel co-ords - x	y		Photo Num	X	y	µm
1	575	296		1	373	170	
2	586	306	8	2	374	162	4
3	598	319	9	3	375	138	13
4	611	335	11	4	376	119	10
5	637	353	17	5	378	81	20
6	LOST FOCUS			6	376	53	15
7				7	381	20	18
8				8	LEFT FRAME		
9				9			
10				10			
Ave distance per second			16	Ave distance per second			19
Cell No:	Cell 8		Distance	Cell No:	Cell 7		Distance
Photo Num	x	y	µm	Photo Num	X	y	µm
1	219	448		1	273	46	
2	231	462	10	2	258	47	8
3	237	478	9	3	243	57	10
4	235	494	9	4	230	65	8
5	226	527	18	5	235	70	4
6	223	548	11	6	220	88	12
7	210	570	14	7	222	112	13
8	OUT OF FRAME			8	229	124	7
9				9	234	133	5
10				10	247	148	11
Ave distance per second			16	Ave distance per second			10
Cell No:	Cell 3		Distance	Cell No:	Cell 6		Distance
Photo Num	x	y	µm	Photo Num	X	y	µm
1	674	327		1	226	410	
2	686	332	7	2	222	419	5
3	697	342	8	3	211	426	7
4	708	354	9	4	202	438	8
5	729	369	14	5	208	453	9
6	739	378	7	6	216	467	9
7	753	396	12	7	221	482	8
8	LEFT FRAME			8	220	501	10
9				9	228	511	7
10				10	221	519	6
Ave distance per second			13	Ave distance per second			10

Table O.2: *Shows the velocity of 8 bacteria for the same sample taken 2 minutes and 7 seconds later.*

Cell No:	5		Distance	Cell No:			Distance
Photo Num	x	Y	µm	Photo Num	X	y	µm
1	240	379		1	403	492	
2	232	375	5	2	403	490	1
3	223	365	7	3	402	492	1
4	212	357	7	4	404	492	1
5	186	351	14	5	403	492	1
6	161	351	13	6	404	491	1
7	LOST CELL			7	405	492	1
8				8	404	493	1
9				9	404	495	1
10				10	403	493	1
Ave distance per second			13	Ave distance per second			1
Cell No:	5		Distance	Cell No:			Distance
Photo Num	x	Y	µm	Photo Num	x	y	µm
1	170	359		1	18	447	
2	177	350	6	2	34	447	8
3	178	337	7	3	48	439	9
4	187	329	6	4	58	433	6
5	195	310	11	5	80	424	13
6	207	292	11	6	104	415	14
7	217	254	21	7	134	378	25
8	223	230	13	8	154	360	14
9	228	207	12	9	174	341	15
10	241	189	12	10	188	318	14
Ave distance per second			14	Ave distance per second			16
Cell No:	3		Distance	Cell No:			Distance
Photo Num	x	Y	µm	Photo Num	x	y	µm
1	309	267		1	521	382	
2	318	258	7	2	539	381	10
3	331	259	7	3	551	388	7
4	344	257	7	4	563	399	9
5	368	255	13	5	576	409	9
6	390	259	12	6	594	421	11
7	429	263	21	7	621	422	14
8	447	255	10	8	640	449	17
9	471	261	13	9	668	441	15
10	497	265	14	10	697	444	15
Ave distance per second			14	Ave distance per second			14

Overall motility measurements, based on comparing subsequent frames, are detailed in Table O.3.

Table O.3: *Motility indices*

Image	Sample 1	Sample 2
	% change	% change
1-2	57.35%	26.23%
2-3	59.63%	26.73%
3-4	63.71%	30.41%
4-5	76.45%	35.65%
5-6	63.73%	33.84%
6-7	71.56%	41.80%
7-8	72.92%	35.80%
8-9	59.37%	33.71%

APPENDIX P: SPORE AND PROTEIN FORMATION

Results for the formation of spores and proteins during the course of the nutrient broth shake flask trial is reported below. The nutrient media trials did not sporulate as their pH became too acidic.

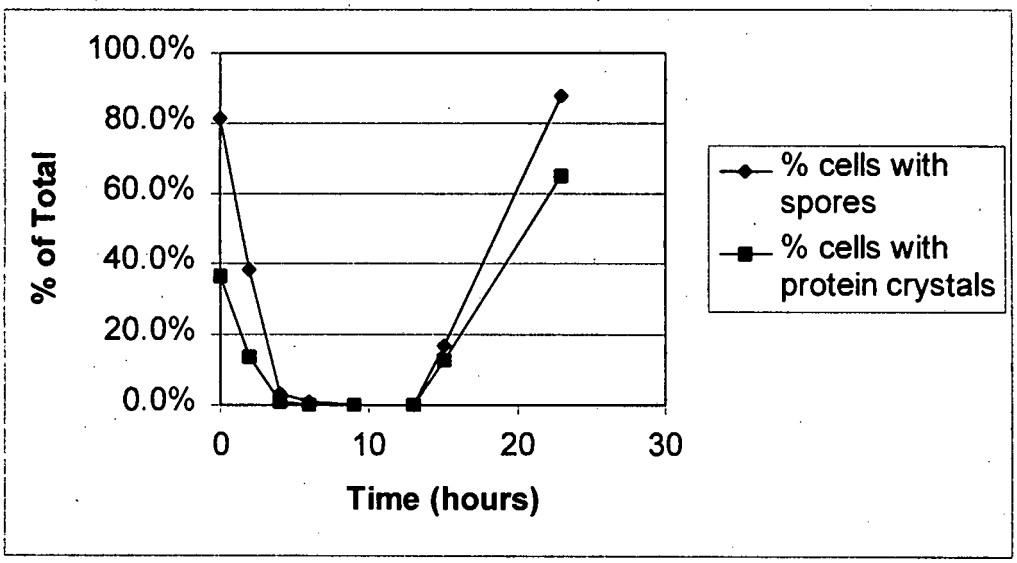


Figure P.1: *Spore and Protein Formation for nutrient broth shake flask trial.*