

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

**Automation of Pollen Analysis using a Computer  
Microscope**

A thesis presented in partial fulfilment of the requirements for the degree  
of

**Master of Engineering**  
in  
**Computer Systems Engineering**

at Massey University, Turitea  
Palmerston North  
New Zealand

Craig Alexander Holdaway

2004

## **Abstract**

The classification and counting of pollen is an important tool in the understanding of processes in agriculture, forestry, medicine and ecology. Current pollen analysis methods are manual, require expert operators, and are time consuming. Significant research has been carried out into the automation of pollen analysis, however that work has mostly been limited to the classification of pollen. This thesis considers the problem of automating the classification and counting of pollen from the image capture stage.

Current pollen analysis methods use expensive and bulky conventional optical microscopes. Using a solid-state image sensor instead of the human eye removes many of the constraints on the design of an optical microscope. Initially the goal was to develop a single lens microscope for imaging pollen. In-depth investigation and experimentation has shown that this is not possible. Instead a computer microscope has been developed which uses only a standard microscope objective and an image sensor to image pollen. The prototype computer microscope produces images of comparable quality to an expensive compound microscope at a tenth of the cost.

A segmentation system has been developed for transforming images of a pollen slide, which contain both pollen and detritus, into images of individual pollen suitable for classification. The segmentation system uses adaptive thresholds and edge detection to isolate the pollen in the images.

The automated pollen analysis system illustrated in this thesis has been used to capture and analyse four pollen taxa with a 96% success rate in identification. Since the image capture and segmentation stages described here do not affect the classification stage it is anticipated that the system is capable of classifying 16 pollen taxa, as demonstrated in earlier research.

## Preface

This research began as an investigation into a number of related fields, gradually gathering structure as knowledge of theory and practicalities of the various fields, particularly optics, was gathered. The aim of the project is to automate the classification and counting of pollen. This thesis is one iteration of the product development cycle. The intention in a single iteration is that it may deliver the intended results, but at worst it will make significant progress towards achieving the aim. I believe that this research has resulted in a useful prototype system for capturing pollen images, and a full system that will serve as a valuable basis for future development.

This research encompasses a range of fields including optics, microscopy, image processing, and programming. As a result it is not expected that the reader will be familiar with all of these fields to a great depth. Therefore the background section of this thesis introduces relevant information in each of the fields. That said, some fundamental background knowledge in the following areas is expected:

- High school level optics, although a little revision is given in the background.
- A general understanding of image processing, as only the specific techniques used in the segmentation are presented in the background.
- Basic programming, ideally with knowledge of C++ and MATLAB™.
- High-school level statistics, mostly for interpretation of the texture formulae, and the mean and variance filter.

A paper detailing the results of the research into capture (sections 3 and 4) was submitted and consequently presented as a poster presentation at the 2003 Image and Vision Computing New Zealand Conference held at Massey University, Palmerston North, in November [1]. This was very well received, and indeed several notable people remarked they while this idea had occurred to them, they were not sure of its virtue or correctness, but are now convinced, and may consider

---

[1] C. A. Holdaway, R. M. Hodgson, "Reinventing the Microscope in the Age of Digital Imaging" in *Image and Vision Computing New Zealand*, 2003, pp.286-290.



applying the idea at their establishments. Additionally a seminar was given to the IIST staff in August detailing the progress made on the image capture research, and a short summary of my research into Wavefront Coding was distributed to the IIST staff. The latter is attached at the appendix of this thesis.

## **Acknowledgements**

My sincerest thanks goes to Professor Bob Hodgson, firstly for providing a project which I felt was both interesting from a research perspective as well as having potential to produce marketable results. Secondly, I would like to thank Professor Hodgson in his supervisory role for his time, patience, knowledge and experience that were all invaluable in completing this thesis. Thanks also to Juliet Newton for her administrative support.

Thank you to Dr. Donald Bailey for his assistance on many matters of image processing. Your expertise was invaluable.

Thank you finally to the Pollen Research Group: Associate Professor David Fountain and Professor John Flenley who provided me with slides and test equipment. Also thanks to Dr. Yongping Zhang for his fantastic work on the classification, Phil Etheridge for his assistance with understanding the classification, and Sophie Cremel for preparing some of the slides.

# Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>10</b>
1.1	OVERVIEW .....	10
1.2	FOCUS .....	11
1.2.1	<i>Scope</i> .....	11
1.2.2	<i>System Overview</i> .....	12
1.3	OUTLINE.....	13
<b>2</b>	<b>BACKGROUND.....</b>	<b>15</b>
2.1	POLLEN .....	15
2.2	OPTICAL MICROSCOPY .....	17
2.2.1	<i>Light and Optics</i> .....	17
2.3	OPTICAL SYSTEMS .....	20
2.3.1	<i>Characteristics</i> .....	20
2.3.2	<i>The Microscope</i> .....	21
2.4	IMAGING.....	23
2.5	IMAGING PROCESSING .....	24
2.5.1	<i>Basic Operators</i> .....	24
2.5.2	<i>Segmentation</i> .....	25
2.6	PRIOR RESEARCH .....	26
<b>3</b>	<b>CAPTURE – ANALYSIS.....</b>	<b>28</b>
3.1	SPECIMEN PREPARATION.....	28
3.1.1	<i>Slide Scanning</i> .....	28
3.1.2	<i>Alternatives</i> .....	29
3.2	ANALYSIS OF CURRENT OPTICAL CAPTURE SYSTEMS.....	30
3.2.1	<i>Confocal Microscope</i> .....	30
3.2.2	<i>Conventional Microscope with a Camera Attachment</i> .....	30
3.2.3	<i>Intel Toy Microscope – QX3</i> .....	31
3.2.4	<i>Wavefront Coding</i> .....	32
3.3	CONSTRAINTS.....	33
3.3.1	<i>Humans</i> .....	33
3.3.2	<i>Breaking away from Human Constraints</i> .....	35
3.4	THE DESIGN OF A SINGLE LENS MICROSCOPE FOR USE WITH AN IMAGE SENSOR.....	37
3.4.1	<i>Basic Parameters</i> .....	37
3.4.2	<i>Early Development</i> .....	38
3.4.3	<i>Experiments to Determine Resolution</i> .....	42

3.4.4	<i>Identifying and Quantifying the sources of Image Degradation</i> .....	47
3.5	CONCLUSION .....	51
3.5.1	<i>Feasibility of a Single Lens Microscope</i> .....	51
3.5.2	<i>Alternatives</i> .....	51
<b>4</b>	<b>IMAGE CAPTURE SYSTEM DESIGN</b> .....	<b>53</b>
4.1	SELECTING AN OBJECTIVE.....	53
4.2	CAMERA REQUIREMENTS .....	60
4.3	SELECTING ILLUMINATION .....	62
4.3.1	<i>Optical Wavelength</i> .....	62
4.3.2	<i>Illumination Source</i> .....	63
4.4	MECHANICAL DISCUSSION .....	65
4.5	FINAL DESIGN OF A COMPUTER MICROSCOPE .....	66
<b>5</b>	<b>SEGMENTATION</b> .....	<b>67</b>
5.1	SEGMENTATION ENVIRONMENT .....	67
5.1.1	<i>Constraints</i> .....	67
5.1.2	<i>Scene Analysis</i> .....	68
5.2	INITIAL APPROACHES .....	71
5.2.1	<i>Basic Methods</i> .....	71
5.2.2	<i>Texture</i> .....	73
5.2.3	<i>Optical Density Sub-sectioning</i> .....	74
5.2.4	<i>Initial Approaches: Summary</i> .....	75
5.3	DEEPER ANALYSIS .....	75
5.3.1	<i>Test Image Statistics</i> .....	75
5.3.2	<i>Mean and Variance Filter</i> .....	75
5.3.3	<i>Edge Statistics and Enhancements</i> .....	78
5.3.4	<i>Region Matching</i> .....	79
5.4	FINAL ALGORITHM .....	81
<b>6</b>	<b>INTEGRATION</b> .....	<b>84</b>
6.1	ANALYSIS OF OPERATOR INTERACTION .....	84
6.1.1	<i>Full Automation</i> .....	84
6.1.2	<i>Calibration by Example</i> .....	85
6.1.3	<i>Summary</i> .....	86
6.2	CLASSIFICATION .....	86
6.3	INTEGRATION .....	87
6.3.1	<i>Testing</i> .....	88

6.3.2	<i>Results</i>	89
6.4	CONCLUSIONS AND RECOMMENDATIONS	91
6.4.1	<i>Capture</i>	91
6.4.2	<i>Segmentation</i>	92
6.4.3	<i>Classification</i>	93
6.4.4	<i>User Interface</i>	93
6.4.5	<i>Conclusion</i>	94
6.5	REFERENCES	95
7	APPENDICES	98
7.1	SOURCE CODE	98
7.1.1	<i>Basic Methods</i>	98
7.1.2	<i>Texture Filtering</i>	98
7.1.3	<i>Optical Density</i>	105
7.1.4	<i>Mean and Variance Filter</i>	108
7.1.5	<i>Edge Detection</i>	108
7.1.6	<i>Complete Segmentation Algorithm</i>	109
7.1.7	<i>Training and Testing Systems (v2.01)</i>	111
7.1.8	<i>Complete Integration Algorithm (v2.02)</i>	115
7.2	REINVENTING THE MICROSCOPE IN THE AGE OF DIGITAL IMAGING	125
7.3	AN INTRODUCTION TO WAVEFRONT CODING	130
7.4	BIBLIOGRAPHY	133

## List of Figures

Figure 1 System Overview .....	13
Figure 2 Scanning Electron Microscope images of a Grass Pollen grain (left) and an Oak Pollen grain (right) [] .	15
Figure 3 Compound Optical Microscope Images, with the background removed, of (left to right) <i>Alopecurus pratensis</i> , <i>Alnus</i> , <i>Cedrus</i> , <i>Acacia dealbata</i> .....	16
Figure 4 Simple Plano-Convex Lens System.....	18
Figure 5 Depth of Field.....	21
Figure 6 Slide Scanning Example .....	28
Figure 7 Intel Toy Computer Microscope QX3.....	31
Figure 8 Olympus BX2-51 in use (left) and cutaway (right) [] .....	34
Figure 9 A square from an England Finder.....	38
Figure 10 Distortion Test Setup .....	39
Figure 11 The effect of applying a narrow-band filter.....	40
Figure 12 Experimental Setup.....	42
Figure 13 Sections (300 × 300 pixels) of Images Captured at 20x Magnification .....	44
Figure 14 Sections (300 × 300 pixels) of Images Captured at 16x Magnification .....	45
Figure 15 Sections (300 × 300 pixels) of Images Captured at 13x Magnification .....	46
Figure 16 Microscope Contrast Comparison. Left: 20x Setup. Right: Compound Microscope 40x.....	47
Figure 17 Spherical Aberration (s.a.).....	48
Figure 18 Experimental Setup.....	53
Figure 19 Sections(300 × 300 pixels) of images captured with a 10x Objective setup at a magnification of 12x.....	54
Figure 20 Sections(300 × 300 pixels) of images captured with a 40x Objective setup at a magnification of 46.5x...	55
Figure 21 Sections(300 × 300 pixels) of images captured with a 10x Objective setup at a magnification of 15x.....	56
Figure 22 Sections(300 × 300 pixels) of images captured with a 10x Objective setup at a magnification of 22x.....	57
Figure 23 Sections(300 × 300 pixels) of images captured with a 40x Objective setup at a magnification of 27x.....	58
Figure 24 Comparative Image Quality of images at different magnifications. All the images are scaled to 12x magnification. ....	59
Figure 25 Image captured at different optical magnifications having the same system magnification. ....	61
Figure 26 Spectral Response of the CCD in the M1024 camera .....	63
Figure 27 Design of a Computer Microscope to optically magnify 12× .....	66
Figure 28 Example images of Background and a Pollen Grain .....	69
Figure 29 Example images of Pollen Grain Clusters .....	69
Figure 30 Example images of Optical Effects.....	70
Figure 31 Example images of Debris.....	71
Figure 32 Three Pollen Types used in the test set (left to right): <i>Fabaceae Sophora</i> (2 pollen grains), <i>Magnoliaceae Michelia doltsopa</i> (2 pollen grains), <i>Lamiaceae Salvia</i> .....	71
Figure 33 Cross-section of background.....	76

Figure 34 Cross-section through a pollen grain.....	77
Figure 35 (left to right) Original image; image with mean and variance filter applied.....	80
Figure 36 (left to right) image with edge detection applied; consolidated edge detection image .....	80
Figure 37 (left to right) image after the AND operation; image after consolidation and removal of small regions. .	81
Figure 38 Segmentation Algorithm.....	82
Figure 39 Image four from the development set: The final result with two pollen grains successfully identified, two large non-pollen objects rejected, and one pollen grain incorrectly identified. ....	83
Figure 40 Classification Sub-System .....	87
Figure 41 Automatic Pollen Classification System .....	88
Figure 42 <i>Betula pendula</i> .....	89
Figure 43 <i>Cyathea dealbata</i> .....	89
Figure 44 <i>Pinus radiata</i> .....	90
Figure 45 <i>Rumex obtusifolius</i> .....	90

# 1 Introduction

## 1.1 Overview

This thesis is a component of the research conducted by the Pollen Research Group at Massey University [2]. An aspiration of this group is to create a system that automatically counts the number of each type of pollen present at a site. Such a system would take appropriately prepared pollen samples, capture and process images of the pollen, and produce the pollen statistics. The aim of this thesis is to create a system that will do this.

Pollen analysis, Palynology, is widely used in reconstructions for archaeology and palaeoecology, has commercial significance in the search for hydrocarbons and is of importance in medical studies [3]. It is also useful in fields that are critical in the New Zealand economy including agriculture, forestry and bee keeping. Palynology is however subject to several disadvantages:

- It is time-consuming because preparation of pollen slides is manual
- A high level of training is needed to obtain accurate identification
- Identification at lower levels (i.e. within a genus) often requires laborious measurement

The automation of pollen identification and counting has the prospect of providing the fine resolution, speed and objectivity that manual pollen identification cannot [4].

The techniques for automating pollen identification suggested in the literature are to use Scanning Electron Microscopy (SEM) [5] and, more recently, Optical Microscopy [6].

---

[2] Pollen Research Group, c/o David Fountain, Institute of Molecular Biosciences, Massey University, Palmerston North, New Zealand

[3] P. Li, J. R. Flenley, L. K. Empson, "Classification of 13 types of New Zealand pollen patterns using neural networks," in *Proc. International Conference on Image and Vision Computing*, Auckland, 1998, pp. 120-123.

[4] E. C. Stillman, J. R. Flenley, "The Needs and Prospects for Automation in Palynology," *Quaternary Science Reviews*, vol. 15, pp. 1-5, 1996.

[5] W. J. Treloar, *Digital Image Processing Techniques and their Application to the Automation of Palynology*, PhD Thesis, University of Hull, Hull, Great Britain.



SEM has the advantage of being able to display the entire pollen grain in focus, however using SEM is expensive and time consuming. Optical Microscopy, while unable to produce in focus images of an entire pollen grain in a single image, has the advantage of being cheaper, faster and more practical for automation. Other means of pollen discrimination, such as DNA analysis, are (for the moment) impractical as they are too time consuming.

## **1.2 Focus**

### **1.2.1 Scope**

The overview has already placed some boundaries on the scope of the project encompassed by this thesis.

#### *1.2.1.1 Boundaries*

The input to the system has been defined as ‘appropriately prepared pollen slides’. These are prepared using techniques outside the scope of this thesis, but in a repeatable manner so that results from different samples can be compared. The needs of the system, particularly the pollen classification stage, determine which slide preparation technique should be used. Equilibrium Density Gradient Centrifugation is such a technique [7].

The output from the system will contain a list of the types of pollen found and a count of how many of each type were found.

#### *1.2.1.2 Constraints*

The system will use optical microscopy, as it is believed to be more economic than other forms of microscopy. The analysis (Section 3) further describes the costs and benefits of the different forms of microscopy. This thesis will not consider in detail the most appropriate method for presenting the pollen to the system or how that can be automated, although a brief discussion of the possibilities is included in section 3.1. Finally, a secondary goal of this project is to create a

---

[6] W. J. Treloar, *Automation of Palynology using Image Processing and Pattern Recognition Techniques*, Postdoctoral Research Report, Massey University, 1994

[7] M. Forster, J. Flenley, “Pollen Purification and Fractionation by Equilibrium Density Gradient Centrifugation,” *Palynology*, 17(1993), pp137-155

simpler microscope by designing it only for computer use, eliminating the constraints placed on optical microscopes by human needs. This thesis explores the hypothesis that removing these constraints will result in significantly reduced cost and give the resulting microscope design significant commercial potential.

### **1.2.2 System Overview**

The use of image processing as the tool of automation dictates the basic structure of the system.

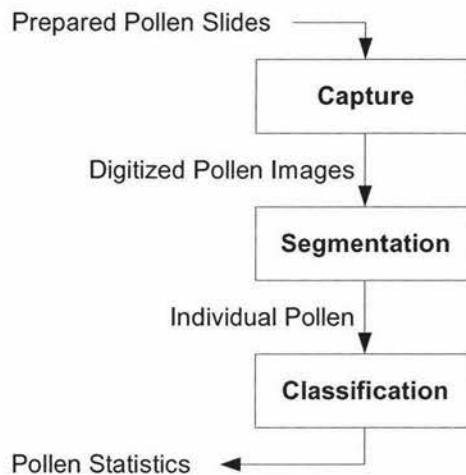
In order to process an image there must first be an image, therefore the first sub-system is defined, called *capture*, in which the image of a pollen sample will be acquired. This subsystem includes the physical hardware used to capture the image, such as a camera, and the software that digitises the image and converts it into a format suitable for processing.

A raw image contains a great deal of information, however most of it is not directly of interest. The second step, *segmentation*, involves isolating the relevant regions within the image, namely the pollen grains, and preparing them in a suitable form for the final stage.

The final stage, *classification*, involves extracting the necessary information from the image and manipulating into the form dictated by the output requirements (Section 1.2.1). This requires the classification of the isolated pollen grains so that they can be counted. This step has been researched and implemented by Zhang[8].

---

[8] Y. Zhang, *Pollen Discrimination Using Image Analysis*, Postdoctoral Research Report, Massey University, New Zealand, 2003



**Figure 1 System Overview**

### **1.3 Outline**

This thesis is structured as a combination of a traditional research-focused thesis and a computer-systems development project.

Chapter 2 provides the background to some of the more advanced concepts used later in the thesis. Chapter 3 analyses the conventional optical microscope and considers extensive possibilities for simplifying the basic design of a microscope. Chapter 4 selects the most feasible of these designs and determines the specifications required of the components within that design. Chapter 5 examines the segmentation subsystem. Rather than try to define an arbitrary break between the analysis and design of the segmentation block, they are combined into a continuous chapter that describes the evolution of the segmentation algorithm. Chapter 6 documents the integration of the capture, segmentation and classification stages, and discusses the results of the integration relative to the design expectations. The remainder of Chapter 6 concludes the thesis by looking at the results of each stage, the results of the full system, and makes recommendations for the future development of a system to automate the classification and counting of pollen.

Much of the project has been implemented in C++ or MATLAB and the source code written for this project can be found in the first appendix. A paper presented at IVCNZ 2003 [1] and

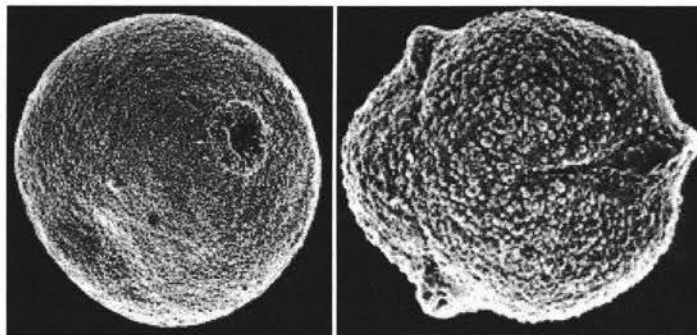
tutorial notes on Wavefront Coding assembled by the author are also attached as appendices. The full working source code, sample images, a copy of this thesis and some relevant specifications can be found on the CD attached to this thesis.

## 2 Background

Frequently in engineering problems, it is necessary to bring together knowledge from different and disparate fields. This section of the thesis briefly outlines some important facets of the fields relevant to this thesis. For a more detailed understanding of the fields discussed it is suggested that you refer to the references provided.

### 2.1 Pollen

A pollen grain is the structure used to transport the male DNA to the female part of a flower. Pollen grains are microscopic, usually about 15 to 100 microns in extent, and just a pinch of pollen powder contains many thousands of grains [9].



**Figure 2 Scanning Electron Microscope images of a Grass Pollen grain (left) and an Oak Pollen grain (right)**  
[9]

Each pollen type has a set of characteristics representative of its family, genus or species, which implies that the species or plant family can be identified. These characteristics are most evident in the outer wall, called the exine, which serves as part of the means for receptive stigmas to recognize compatible pollen. The exine is composed of an extremely stable material called sporopollenin. This material is so stable that pollen grains many thousands of years old retain the same texture and pattern [10]. The pattern is to some extent determined by the number and position of its germinal apertures: Spherical grains tend to have one or several o-shaped apertures, long grains usually have three or more apertures, and triangular grains three

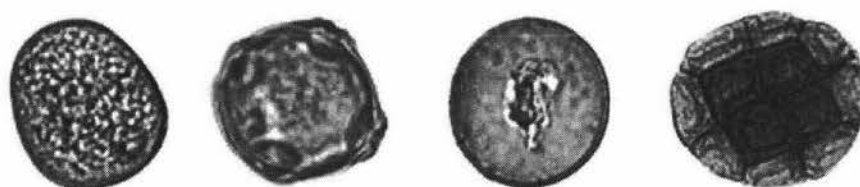
---

[9] J. Emberlin , B. Adams-Groom , "What is Pollen," <http://pollenuk.worc.ac.uk/aero/pm/WIP.htm>

[10] T. E. Weier, M. G. Barbour, C. R. Stocking, T. L. Rost, *Botany*, Singapore: John Wiley & Sons, 1982, pp286-295

apertures[11]. Apertures may be long furrows, round pores or a combination of the two structures.

Detailed pollen images are obtained using scanning electron microscopy, two examples of which are shown in Figure 2. When lower magnification optical microscopy is used both the three-dimensional nature and the detail of the pollen is lost. The images obtained using transmission optical microscopy are a combination of the pattern of the internal contents and the walls facing the microscope, and the walls on the edges of the pollen (from the microscope viewers' perspective), which appear more defined. Figure 3 shows four images of pollen grains captured using optical microscopy.



**Figure 3 Compound Optical Microscope Images, with the background removed, of (left to right) *Alopecurus pratensis*, *Alnus*, *Cedrus*, *Acacia dealbata*.**

Pollen grains are developed in the anthers of a flower. Upon readiness, the pollen is released to be spread, usually by the wind or by insects. These two chief forms of dispersal result in different pollen characteristics, for instance pollen dispersed by the wind tends to be smooth and non-adhesive, while pollen dispersed by insects involves very specific co-adaptation such as adhesives to ensure the pollen will stick to the insect [11].

#### *2.1.1.1 Applications*

As a result of the durability of the exine, ancient pollen can be identified based on the exine characteristics, even when all other traces of biological material are lost [10]. The reconstruction of past vegetation types and climates by the use of fossil evidence is called Paleoecology.

Pollen causes allergic reactions in some humans, causing widespread medical conditions such as hay fever and asthma. Gaining insight into the processes of pollen dispersal and human reaction

---

[11] R. B. Knox, *Pollen and Allergy*, Southampton: Edward Arnold Publishers, 1979, pp3-8, 22-24

to pollen contributes to the understanding of the health effects of pollen. Significant research is being undertaken in this area [12].

As part of the reproductive system in plants, pollen grains are vital in horticulture and forestry, both of which are large industries in New Zealand. A smaller industry, bee keeping, benefits from pollen in a more direct manner.

## **2.2 Optical Microscopy**

### **2.2.1 Light and Optics**

Visible light is a band within the electromagnetic spectrum, with a wavelength ranging from about 380nm (violet) to 700nm (red). The propagation of light can be represented by two types of models: geometric and wave optics [13].

#### *2.2.1.1 Geometric Optics*

Light does not travel at a universally constant velocity. The speed of light depends on the material through which the light is passing, called the optical medium. In more optically dense media, light travels more slowly. Therefore when light passes from one medium to another it changes velocity. This change in velocity causes the direction of the light to bend, a phenomenon called refraction. Geometric optics models the transmission, reflection and refraction of light using rays.

A lens is typically made up of an optically dense material containing two or more refracting surfaces at least one of which is curved. Lenses may be used in an optical system to modify a beam of light or to form an image of an object. There are a number of factors that need to be considered when characterizing a lens:

- **Diameter:** The diameter of a lens, often restricted by an aperture, is typically chosen based on the size of the light beam and object that needs to be imaged. It is important for controlling the depth of field, diffractive effects, and aberrations.

---

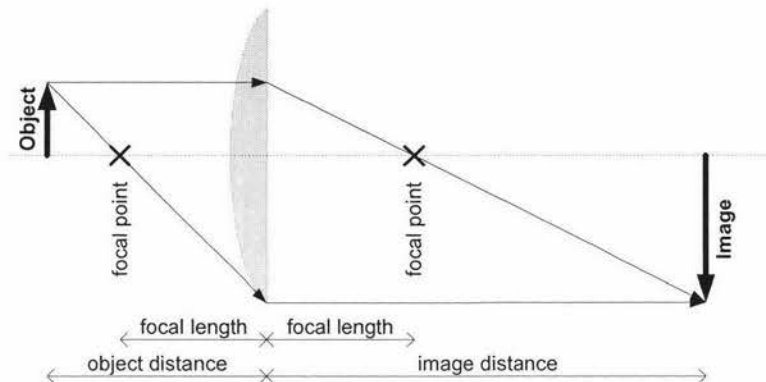
[12] D. W. Fountain, "Pollen and Inhalant Allergy," *Biologist*, vol. 49, no. 1, 2002, pp.5-9

[13] H. D. Young, R. A. Freedman, *University Physics*, Reading, Mass: Addison-Wesley, 1996, pp.1053-1192

- Focal length: The focal point is defined as the point at which parallel rays coming into the lens converge. The distance between the centre of the lens and this point is the focal length  $f$  of the lens.

The only type of lens relevant to this thesis is the plano-convex lens. This is a lens with a single spherical surface and an optically flat surface opposite, as shown (sliced to two-dimensions) in Figure 4.

An object can be represented by its principle rays, shown in Figure 4. The point where these rays converge is called the image. The placement of the object with respect to the lens determines the properties of the image such as its magnification and its orientation.



**Figure 4 Simple Plano-Convex Lens System**

#### 2.2.1.2 Limitations – Wavelength and Diffraction

The wavelength of light imposes the key limitation on the resolving power of optical microscopy, because any object that is smaller than the wavelength of light cannot be resolved. Thus the smallest object that can be seen using standard optical microscopy has a size of about 380nm, the wavelength of violet light.

When a wavefront passes through an aperture, the light at the edge of that aperture is diffracted. For example, assuming a 'perfect' lens, geometric optics determines that a point object will converge to a point image. However, in practice the image formed has a bright central point surrounded by a series of concentric rings of rapidly decreasing brightness due to diffraction



[14]. An image is made up of many points of varying intensity, and the additive effect of the diffraction of all the points causes blurring.

#### 2.2.1.3 Limitations – Aberrations

Thus a ‘perfect’ optical system is one which is ‘diffraction limited’. However lenses suffer from imperfections due to their shape and physical properties. These imperfections result in aberrations. A hyperbolic lens can theoretically be crafted to eliminate the aberrations caused by shape, however it is very difficult to grind a hyperboloid, but easy to grind a spherical surface. Consequently most lenses have spherical surfaces, and therefore suffer from aberrations.

Spherical Aberration is the result of imperfections in the lens that cause different sections of the lens to focus at different focal lengths. Generally an annular section of the lens, which has some radius from the lens centre, will focus to one point. Because each annulus focuses to a different point the lens exhibits spherical aberration. Spherical aberration causes the image to blur.

Curvature of field is a result of imaging with a flat image plane when the wavefronts of the light from the lens actually form a curved surface. Pincushion and barrel distortion occur because the magnification near the edges of the lens is different to the magnification near the centre.

The refractive index of a material, a measure of optical density, varies with the wavelength of light passed through it, resulting in violet light rays bending further than red rays. This causes ‘rainbow edges’ in colour images, and blurring in monochromatic images. This is known as chromatic aberration.

To gain a more detailed understanding of all aspects of optics see Pedrotti and Pedrotti’s *Introduction to Optics* [15].

---

[14] B. Walker, *Optical Engineering Fundamentals*, McGraw-Hill Inc., 1995, pp.50-53

[15] F. Pedrotti, L. Pedrotti, *Introduction to Optics*, Englewood Cliffs, N.J.: Prentice-Hall International, 1993

## 2.3 Optical Systems

### 2.3.1 Characteristics

#### 2.3.1.1 Magnification

The magnification of an optical system is the ratio of image size to object size. In microscopy there is the concept of *empty magnification* where an increase in the magnifying power of a lens system does not result in any greater detail in the image, as the ultimate image resolution has been reached at a lower magnification.

It is important in this research to note the distinction between *Optical Magnification* and *System Magnification*:

- Optical magnification is the enlargement of the object achieved by the lens alone.
- System, or total, magnification is the enlargement between the object and the final displayed or reproduced image.

In a conventional optical microscope, all of the magnification is achieved optically and therefore the optical magnification is equal to the system magnification. However in a computer microscope some of the magnification is achieved as a consequence of the size difference between the image sensor and the display, called Pixel Scaling (section 3.2.3), and therefore the optical and system magnification are different.

#### 2.3.1.2 Depth of Field

The Depth of Field of an optical system (imager and lens) is the range of distances from the lens for which the image of an object is 'in-focus'. An object is considered to be in-focus if the image of a point object is smaller than the resolution of the imaging device.

The Depth of Field can be calculated using the formula presented here [16]:

$$D.o.F. = \frac{2Ads_o(s_o - f)f^2}{f^4 - A^2d^2s_o^2} \quad (2.1)$$

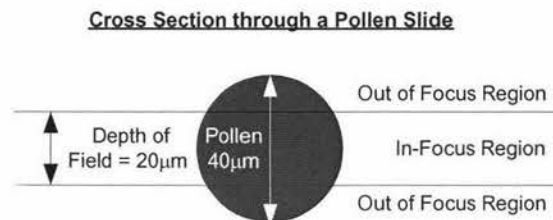
---

[16] F. Pedrotti, L. Pedrotti, *Introduction to Optics*, Englewood Cliffs, N.J.: Prentice-Hall International, 1993, pp.126-129

Where:

- $f$  is the focal length of the lens.
- $A$  is the aperture variable, calculated from  $A = \frac{f}{D}$  where  $D$  is the aperture diameter.
- $s_o$  is the distance from the lens to the object.
- $d$  is the blurring factor. This is a measure of the resolution of the image sensor, and should therefore be set to the size of a CCD element for digital camera calculations.

The depth of field for a compound microscope is in the order of  $1\mu\text{m}$  to  $10\mu\text{m}$ . Given the typical size of, say pollen, at  $50\mu\text{m}$ , the entire pollen cannot be in focus at the same time. Consequently any images of pollen contain an in-focus slice of pollen, mixed with regions of pollen that are out of focus. Figure 5 illustrates this concept of depth in microscopy.



**Figure 5 Depth of Field**

### 2.3.2 The Microscope

In its broadest definition a microscope is some optical device that makes objects visible that cannot be seen with the naked eye. In order to view pollen grains optically we need to use a microscope with a total magnification between  $400\times$  and  $1000\times$ .

#### 2.3.2.1 Simple Microscope

At its simplest, a single lens can constitute a microscope. For instance the single lens setup in Figure 4 will make the image twice the size of the object enabling us to see parts of the object which were too small to resolve unmagnified.

There are limits to the magnification that can be achieved with a single lens. These limits are due to the effects of diffraction and aberrations. Typically chromatic aberration is the first noticeable degradation, and limits simple uncorrected magnification to about  $5\times$  [17].

#### 2.3.2.2 Compound Microscope

Invented around 1590, a compound microscope has an objective lens (the lens closest to the specimen being viewed) and an eyepiece so mounted that they may conveniently be moved relative to the specimen to focus the image [18]. The magnification produced by a compound microscope is the product of the magnification of the lenses making up the eyepiece and the objective. Standard microscope objectives have magnifications of the values  $4\times$ ,  $10\times$ ,  $40\times$  and  $100\times$ . Eyepieces generally magnify  $10\times$ .

To achieve the magnification required to view pollen, a combination of lenses are needed at both the objective and the eyepiece. At these magnifications aberrations (Section 2.2.1.3) cause degradation to the image, and additional lenses are required to correct for these. The corrective lenses are primarily located on the objective.

Lighting becomes more critical as the magnification increases. At high magnifications ( $>40\times$ ) Köhler illumination is used. Köhler illumination consists of a train of lenses and irises, where the final lens, the condenser, provides parallel and evenly distributed illumination of the specimen [19].

An example of a current fully equipped compound microscope is the BX2-51RP, which has over 25 lenses, spaces for filters, a rotating slide table, and infinity corrected achromatic optics, costs

---

[17] B. H. Walker, "7.3 A Typical Lens Application," in *Optical Engineering Fundamentals*, McGraw-Hill, 1995, pp.150-154.

[18] "Microscope", *Encyclopaedia Britannica*. Encyclopaedia Britannica Premium Service. 2 Mar 2003. [www.britannica.com](http://www.britannica.com)

[19] E. M. Slayter, H. S. Slayter, "The Light Microscope," in *Light and Electron Microscopy*, Cambridge University Press, 1992, p.131-148

US\$16,853.00 and is shown later in this thesis in Figure 8 [20]. This price does not include cost of a camera for capturing images.

#### *2.3.2.3 Advanced Techniques*

The resolution of even the most expensive compound microscopes is limited by the information that can be transmitted by light. There are however techniques for 'extending' the resolution of optical microscopes beyond the theoretical limit of light resolution. The more significant of these are described below.

### **Fluorescence**

In fluorescence microscopy, near ultra-violet radiation is used to excite fluorescent material which emits light of a visible wavelength, allowing objects to be seen that have sizes of the wavelength of near UV light (200-400nm) [19].

### **Confocal**

Confocal microscopy uses a laser beam to image very narrow slices of a specimen. These slices are then reconstructed into a 3-D image of the specimen. Confocal microscopy offers several advantages over conventional optical microscopy, including shallow depth of field, elimination of out-of-focus glare, and the ability to collect serial optical sections from thick specimens [21]. However, the equipment required is even more expensive than current compound microscopes, and the capture process is time intensive.

## **2.4 Imaging**

A silicon based digital image sensor is an array of elements, each of which converts incident photons (light) into electrons. The number of electrons produced represents the total intensity of the light incident over that element integrated for the time it was exposed. The quantity of electrons at all the elements in the array can be read electronically, producing a digital representation of the light falling on the sensor, otherwise known as a digital image.

---

[20] McCrone Microscopes, "Olympus BX2-51RP (Transmitted/Reflected)," [www.mccrone.com](http://www.mccrone.com)

[21] S. W. Paddock, T. J. Fellers, M. W. Davidson, "Introduction to Confocal Microscopy" [www.microscopyu.com/articles/confocal/confocalintrobasics.html](http://www.microscopyu.com/articles/confocal/confocalintrobasics.html)

Digital image sensors come in two main varieties. Charged Coupled Devices, CCDs, and Complementary Metal Oxide Semiconductor sensors, CMOS. Typical image sensors used today range in size from 0.3 to 8 megapixels. Colour imaging is usually achieved by applying a colour filter over the sensor, such as a Bayer Tile. As a consequence of their specialized manufacturing process, CCDs tend to be used in cameras that focus on high-quality images with lots of pixels and excellent light sensitivity. CMOS sensors, manufactured in the same manner as most integrated circuits, traditionally have lower quality, lower resolution and lower sensitivity. CMOS sensors are just now improving to the point where they reach near parity with CCD devices in some applications [22].

## **2.5 Imaging Processing**

### **2.5.1 Basic Operators**

The lowest level or first stage of image processing, sometimes referred to as pre-processing, involves the enhancement of relevant information in an image and the suppression of irrelevant information

#### *2.5.1.1 Arithmetic and Logic Operators*

All of the traditional arithmetic (+, -,  $\times$ ,  $\div$ ) and logic (AND, OR, NOT) operators have image pre-processing equivalents. These are point operators, treating each pixel or corresponding pair of pixels independently throughout an image or multiple images.

These operators are constrained by the number of levels in a greyscale image. For instance in subtracting one image from another, some pair of pixels may introduce the subtraction: 45-60. The result arithmetically is -15, but this is cannot be represented in a greyscale image where the limits are integers in the range [0,255]. To accommodate this and similar underflows and overflows from other arithmetic operations special rules must be applied. In the subtraction

---

[22] How Stuff Works, "What is the difference between CCD and CMOS image sensors in a digital camera?," Jan 2004. <http://www.howstuffworks.com/question362.htm>

example, the result: could be wrapped around to be  $256-15 = 241$ ; could be made absolute to equal 15; or saturated to 0.

Logical operators act as would be expected on binary images, however on greyscale images the operator is not well defined. For instance the result of taking the inclusive OR of two pixels could add their values, or take the binary representation of the pixel intensities and logically OR the binary values (the bitwise OR), or something else altogether.

#### *2.5.1.2 Morphological Operators*

Morphological operators perform actions that alter the form and structure of regions within images. Morphological operators are simplest to understand when applied to binary images, but are equally applicable in greyscale images. Morphological operators are performed by convolving a small matrix, called the structuring element, with the image. The two key operators are erosion and dilation. As is implied by the names, erosion removes pixels from the edges of a region and dilation adds pixels to the edges of a region.

More complex morphological operators are mostly a combination of these basic ones. Opening consists of erosion followed by dilation and removes narrow connections between objects in the image. Its opposite, closing, is composed of dilation followed by erosion and bridges small gaps between objects and fills in holes.

### **2.5.2 Segmentation**

Segmentation subdivides an image into its constituent regions or objects. Segmentation should stop when the objects of interest in an application have been isolated [23].

#### *2.5.2.1 Edge Detection*

Edges are the key attribute of regions that enables their extent to be defined. Gradient operators for detecting edges are well developed; the most widely known of these is the Sobel filter consisting of two matrices applied to an image which detect vertical and horizontal edges

---

[23] Gonzalez R. C., Woods R. E., "10 Segmentation" in *Digital Image Processing*, 2<sup>nd</sup> ed., New Jersey: Prentice-Hall, 2001, pp.567-642



independently. A drawback of edge detection is that it suffers from noise. Small amounts of additive noise can introduce errors in a detection system that relies on a small  $3 \times 3$  matrix such as the Sobel operator.

#### 2.5.2.2 Regional Operations

An alternative method of segmentation is to make the distinction between regions based on discontinuities in the grey-levels of an image, usually by taking a threshold of the image. Regional operations are less noise sensitive than edge detection methods.

## 2.6 Prior Research

For many years the automation of pollen analysis has been seen to be desirable and useful [24], but until recently it has not been practical. Stillman and Flenley [4] identified image processing as a tool for automating pollen analysis. In recent years image processing systems have increased in performance, which can be attributed both to improved algorithms and a dramatic increase in available computing power.

As pollen grains can be very difficult to classify visually, successful classification methods have involved the use of texture. Treloar [6] extended the work of his Ph.D. thesis on classifying SEM pollen images to pioneer textural classification of pollen using images captured by an optical microscope. More recently Li and Flenley demonstrated the feasibility of using neural network classifiers. This work achieved 100% classification of 13 pollen types [25]. Following on from that work, Zhang used wavelet features to classify 16 airborne pollen types and to differentiate between airborne and non-airborne pollen [8].

The literature reviewed thus far considers the difficult problem of classification. Achieving the automation of pollen analysis also requires the automation of the other stages of the pollen

---

[24] J. R. Flenley, "The problem of Pollen Recognition", in M. B. Clowes and J. P. Penny, *Problems in Picture Interpretation*, pp141-145. CSIRO, Canberra, 1968

[25] J. R. Flenley, P. Li, L. K. Empson, "Identification of 13 Pollen Types by Neural Network Analysis of Texture Data Only," in *Proc. Of Image and Vision Computing New Zealand*, 1999, pp.295-298



analysis process. Until recently this problem had not been addressed. France et al. [26] demonstrate a system that deals with the process of 'capture to classification'. The hierarchical nature of their system provides a useful means of rejecting detritus, and the techniques used to extract objects and features are successful. This method differentiates between only three pollen types.

---

[26] I. France, A. W. G. Duller, G. A. T. Duller, H. F. Lamb, "A new approach to automated pollen analysis," *Quaternary Science Reviews*, vol.19, 2000, pp.537-546

## 3 Capture – Analysis

This section discusses the alternatives available for a capture system, beginning with a wide ‘all-possibilities’ approach, and gradually selecting alternative components until a complete design has been determined.

### 3.1 Specimen Preparation

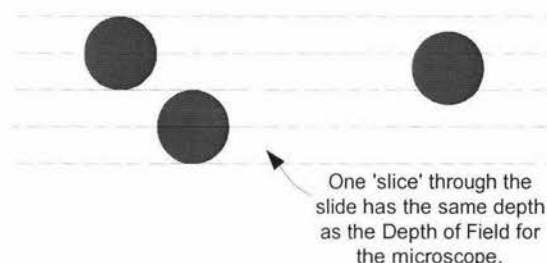
It is a requirement of the capture system that pollen will be present on the slide in the region where the microscope is viewing. How the pollen specimen comes to be where it is needed is almost outside the scope of this thesis and is a subject worth examining only briefly.

#### 3.1.1 Slide Scanning

Currently slides are prepared using specialist techniques and then examined under three-dimensional manual control. The first level of automation on this is to enhance the manual controls with computer-controlled servomotors. These are expensive, but are used on high-end microscopes, and allow both open and closed loop control of the slide position.

##### 3.1.1.1 Scanning a Complete Slide

Capturing an entire slide involves scanning in three dimensions. A slide can be considered to consist of a number of levels of depth, where each level has the same depth as the depth of field of the microscope (see section 2.3.1.2). For instance to scan all three objects in Figure 6, four slices are required. At each level, a plane, the remaining two dimensions, must be captured.



**Figure 6 Slide Scanning Example**

While scanning in all three dimensions is a simple method, it is highly time consuming. This can be improved by scanning only a few depth levels. Objects which are in nearby depth planes

will have a characteristic diffraction blur, and can be investigated by altering the focal depth at that position.

### *3.1.1.2 Statistical Sample*

Scanning the entire slide is time consuming. Sampling is a statistical technique that permits the capturing of a subset of the full slide, which is representative of the contents of the whole slide. Unfortunately not all pollen taxon distribute pollen randomly within a slide. Some pollen tends to cluster by nature, and others tend to attach to debris present in the slide. Additionally sampling makes it less likely that the '1 in 1000 pollen', an aim of Stillman and Flenley [4], will be found.

### *3.1.1.3 Slide Scanning Trade-Offs*

Any technique other than an exhaustive scan of the slide runs the risk of missing pollen. However, as some pollen tend to cluster and can be obscured by both pollen and debris, it is impossible to count every single pollen on a slide reliably.

Thus, given counting every pollen is not possible, it seems sensible to conduct a sample. To obtain a more accurate representation of the pollen population it is important that the slides themselves reflect a random sample of the population.

## **3.1.2 Alternatives**

Breaking away from traditional ideas and using different technology when existing technology has not been satisfactory for the job is a theme of this research. The slide is such a technology. The slide limits the capture speed because:

- It is 3-Dimensional
- It has to be controlled in 3 axes to capture images requiring fine, complicated, and expensive x-y-z controls.

So what alternatives exist to these? The following are only suggestions, and are not founded by anything other than possibility:

- A moving sticky tape that can collect airborne pollen by causing pollen blown onto the tape to adhere. The tape can be moved in front of the capturing device and a line scan system used relieving the two planar axes of control.

- Pollen could be electro-statically charged pulling them onto a single plane, removing the depth dimension.

## **3.2 Analysis of Current Optical Capture Systems**

Analysis of a capture system involves examining existing capture technologies and considering the strengths and weaknesses of each. Typically, constraints on performance, optically and in usability, have led to the design decisions made. A changing environment can change the effect or validity of existing design constraints. If the constraints applied to previous designs are found to no longer be valid, for instance by virtue of improved technology, then there is a potential for an innovative design to give better performance or to reduce cost. There are a variety of optical technologies that can and have been used to view pollen at or near the resolution we require (as detailed in section 2.1). The costs and benefits of each are discussed in the following sections.

### **3.2.1 Confocal Microscope**

Confocal Microscopes, introduced in Section 2.3.2.3, provide the highest existing optical resolution by using fluorescence, with the improvements to image quality provided by the narrow depth of field. Controlling the fine acquisition of image slices requires precise control and multiple image slices to create a single image. Images generated by a confocal microscope contain significantly more depth information than conventional optical microscope images [21]. However, Confocal microscopes are extremely expensive, due primarily to the precision required in their construction and their operation. It also takes considerable time to acquire a single complete image.

### **3.2.2 Conventional Microscope with a Camera Attachment**

Conventional compound microscopes, introduced in section 2.3.2.2, provide the greatest flexibility of current optical microscopes. Attachments for specialized techniques such as dark-field illumination, phase-contrast and polarization are readily available [27].

To attach a camera additional optics are required to transmit the image at the back focal plane to the camera. At low system magnifications, the optical magnification required for the addition of

---

[27] E. M. Slayter, H. S. Slayter, *Light and Electron Microscopy*, Cambridge University Press, 1992

a camera is less than  $1\times$ , so the optics required actually diminishes the image. This is a peculiar design feature of compound microscope with camera attachments, where an objective magnifies the image and further optics then diminish it. This is a significant redundancy in the optical components. However, a professional level compound microscope is expensive, usually over \$10000, and this cost prevents laboratories from having large numbers of the instruments.

### 3.2.3 Intel Toy Microscope – QX3

Unlike regular optical microscopes, the QX3 has no eyepiece, instead it has a built-in camera that sends video images of specimens or small objects at  $10\times$ ,  $60\times$ , or  $200\times$  magnification to a PC via a Universal Serial Bus (USB) connection [28].



**Figure 7 Intel Toy Computer Microscope QX3**

The key to the magnification is not however from high-powered objectives, in fact the optical magnification of the system is  $0.2\times$ ,  $1.1\times$  and  $4\times$  respectively. The key is the ‘pixel scaling’.

---

[28] L. Jelinek, G. Peters, J. Okuley, S. McGowan “Dissection of the Intel® Play™ QX3™ Computer Microscope,” *Intel Technology Journal*, Q4, 2001

Pixel scaling is the ratio between the size of a pixel in the sensor array, and the size of a pixel in the display (it is often easier, and equally valid, to use the size of the entire array and entire display to calculate this).

For example, the pixels on a 15" monitor, running at  $800 \times 600$  resolution, are roughly  $270\mu\text{m} \times 270\mu\text{m}$ . When the  $9\mu\text{m} \times 9\mu\text{m}$  pixels of the image sensor are displayed on the monitor, the sensor pixels are "magnified" 30 times [28].

#### *3.2.3.1 Benefits of the QX3*

The key benefit of the QX3 is its price. The QX3 was designed to cost less than US\$100 [28]. Most microscopes capable of resolving to about  $3\mu\text{m}$  cost more than this, let alone the considerable cost of adding a camera to them.

#### *3.2.3.2 Costs of the QX3*

The QX3 is designed as a toy, and not a scientific instrument, consequently it is too simple for application to pollen analysis. For instance: the specimen table is poorly mounted and cannot be finely adjusted; the image integration time is visibly long (to the order of 10 seconds); and its magnification is not high enough for our purposes.

#### *3.2.3.3 Similar Products: the Olympus MIC-D*

During the period of my research Olympus brought out the MIC-D, which improves on the QX3 by providing a more robust gliding stage and higher optical magnification (up to  $9\times$ ). This is almost the optical magnification required for this research. However the sensor size is quite small at  $640 \times 480$  [29]. I have not been able to obtain this microscope for comparison.

### **3.2.4 Wavefront Coding**

In wavefront coding aspheric optical elements and digital signal processing techniques are used to enhance the performance of imaging systems. Wavefront coding has been applied to imaging systems to extend depth of field and to control aberrations. These applications have demonstrated an image quality at least equivalent to that of a more expensive traditional optical

---

[29] Olympus, "Olympus MIC-D Product Information", <http://www.mic-d.com/product/spec.html>, 12 Jan. 2004.

system, with 4-5× greater field of view, without changing aperture sizes in the system [30]. For further details see the appendix ‘Wavefront Coding’.

An extended depth of field is likely to be useful for classifying pollen as most of the information in the optical images comes from the surfaces of the pollen. Extending the depth of field may allow the structure of the surface to be seen more easily, as the extended depth of field of confocal microscopy has shown [21]. However rather than suffer the expense and long capture time of a confocal microscope, wavefront coding can be applied to existing compound microscopes. I have not had the equipment or the time to experiment with the possibility of using wavefront coding.

### **3.3 Constraints**

There are many optical limitations that constrain the design of an optical microscope. These are fundamental and cannot be avoided regardless of design. Some constraints are placed on the design of a microscope by the characteristics of a specimen slide. For instance a standard slide is one inch wide and three inches long, so the design of a microscope must then provide a slide table large enough to hold this slide, and the x-y controls needed for fine control of the slide. However, none of the above is the most significant constraint on the design of microscopes. The major limitation is the human observer.

#### **3.3.1 Humans**

Microscopes are inherently designed to be used by humans. This observation can be justified by studying the components of a, say, compound microscope as the BX2-51 (quoted in section 2.3.2.2 and illustrated in Figure 8).

##### *3.3.1.1 Optics*

Human eyes contain an integral lens and are not capable of directly viewing microscopic objects and therefore require optics to provide magnifications in the range 400× – 1000× required to view pollen grains. To obtain magnification of this magnitude and preserve image quality

---

[30] S. C. Tucker, W. T. Cathey, E. R. Dowski, “Extended depth of field and aberration control for inexpensive digital microscope systems,” *Optics Express*, vol. 4, no. 11, pp. 467-474, 24 May 1999



requires complex optics, as simple optics designed to provide high magnification alone will exhibit severe image degradation due to lens aberrations.

#### *3.3.1.2 Illumination*

Human eyes require a suitably intense and invariable level of illumination. This is achieved in a microscope by the illumination train and the condenser. The illumination can be finely adjusted by closing an iris under the condenser.

#### *3.3.1.3 Shape*

Traditional microscopes have been designed to facilitate focusing by means of a knob on the arm of the microscope. This is performed while the user is looking through the eyepieces. Consequently arm length determines the distance that can be used, limiting the optical tube length and constraining the optical design. This is illustrated in Figure 8 (left)

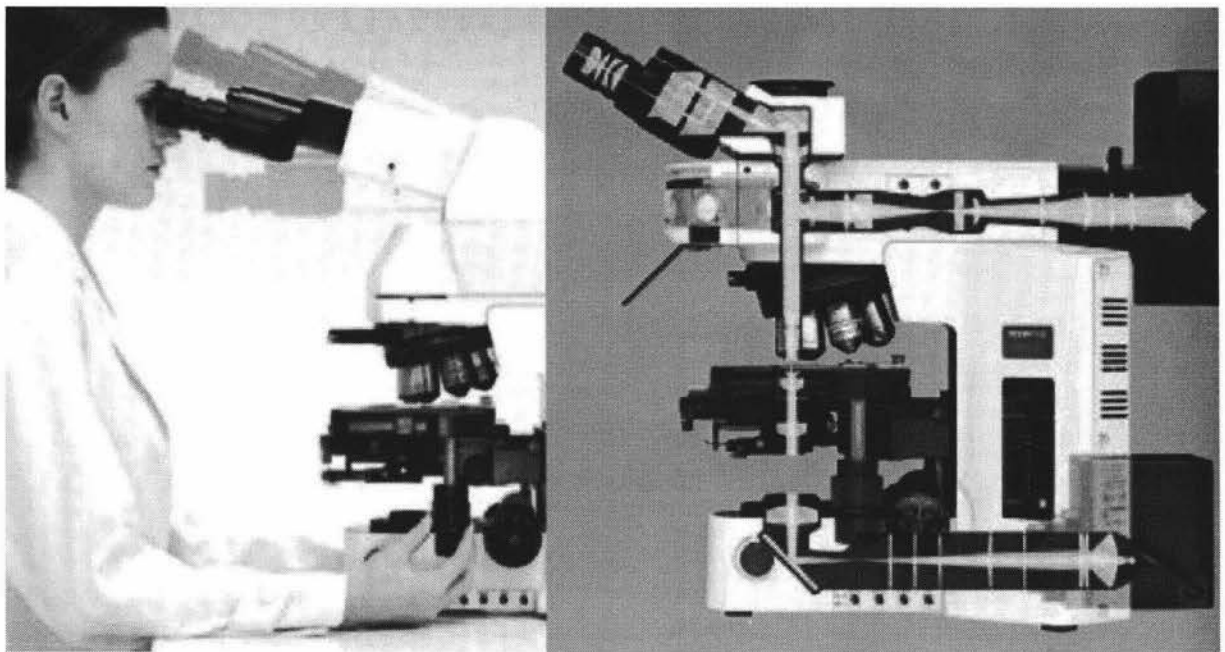


Figure 8 Olympus BX2-51 in use (left) and cutaway (right) [31]

---

[31] Olympus Corp., “BX51 Research System Microscope Brochure”, [www.olympus.com](http://www.olympus.com), January 2004



### 3.3.2 Breaking away from Human Constraints

#### 3.3.2.1 Eyes

Human eyes are the primary constraint on the design of a microscope. They dictate the magnification, illumination and aberration correction required. Consider the effect of human eyes on the optics: A typical laboratory microscope, such as that illustrated in Figure 8, suitable for viewing pollen has 8-15 lenses in the objective and 2-5 lenses in each eyepiece. Given that every lens added to the microscope adds expense, it is logical to seek a means of replacing human eyes with a technology that does not require such high magnification, and is more suitable for reliably analysing and classifying microscopic objects.

#### 3.3.2.2 A Microscopic Sensor

Of the microscopes analysed in section 3.2, the Intel QX3, at US\$100, was the least expensive by about two orders of magnitude. This is because it requires only a single lens with low magnification imaging onto an image sensor and pixel scaling provides the rest of the magnification.

An image sensor, such as a CCD, is microscopic in nature. The size of an element, a pixel, on the sensor is similar in magnitude to the microscopic detail that is of interest. This substitution of a digital sensor for the human eye affects the design of the optics:

- The required magnification is greatly reduced.
- Reduced magnification reduces the number of lenses required to enlarge the object.
- Reduced magnification lessens the effect of aberrations, reducing the number of lenses required for aberration correction.

Thus significant savings in optics, and therefore cost, can be made by removing from the microscope the optics that are required to meet the needs of human eyes.

#### 3.3.2.3 Further Benefits of Using a Digital Image Sensor

The benefits of replacing human eyes with an image sensor are not only a reduction in cost. By using an image sensor the design is freed of the illumination and shape constraints discussed above. This substitution provides additional benefits.

## **Illumination**

The dynamic range of an optical instrument is the ratio between the maximum and minimum intensities that the instrument can detect and process. Although an image sensor has a much smaller dynamic range than our eyes, the exposure range can be 'shifted' by altering the integration time of the sensor (the time over which the sensor is exposed to light) to suit the intensity of the scene. This allows for illumination by sources too intense for human eyes, such as a laser.

Furthermore, human eyes exhibit adaptation, where over a period of time measurable in minutes, human eyes become accustomed to a significantly changed illumination level [32]. Unlike human eyes, a digital image sensor does not need time to respond to sudden changes in illumination level. This allows flashed lighting to be used, where the sensor is synchronized to capture as the flash illuminates the specimen. Human eyes, due to their adaptation time and range cannot cope with either intense or variable illumination.

## **Shape**

Once an image sensor is introduced, human physiology no longer dictates the distance between the object and the imaging sensor. This allows optical tubes of any length. Also, the shift from a binocular head to a monocular head immediately reduces the number of lenses required if an eyepiece is required at all. It is important to note there are benefits to using binocular viewing, such as improved depth perception, which are lost using a monocular eyepiece.

### *3.3.2.4 Microscope Output*

Replacing human eyes with an image sensor does not remove the need for human interaction. Humans are required to focus the microscope, to position the slide using fine x-y controls, and humans have to see the final image results for analysis. For these requirements to be met, the images captured by the image sensor must be displayed to the human operator in real-time.

---

[32] T. N. Cornsweet, *Visual Perception*, New York: Academic Press Inc., 1970. pp.7-9

This implies the need for a display. Different displays have different properties, such as the pixel size and screen resolution. A variation in these properties will result in different magnifications due to pixel scaling, and interpolation of images by some displays. This variation will produce inconsistent images, and may introduce empty magnification. This inconsistency in image production is not a severe disadvantage, but as with conventional microscope usage, the user must be aware of the potential to introduce empty magnification by, in computing terms, zooming in too far.

The need for a computer is a constraint on a digital system. However given the widespread usage and inexpensiveness of computers, it is not considered to be a problem, rather it represents necessary infrastructure for computer microscopes, which is already in place.

### **3.4 The Design of a Single Lens Microscope for use with an Image Sensor**

The ultimate in minimization of a microscope would be to reduce it to a single lens. If this could be done, it would vastly simplify the construction and so reduce the cost.

#### **3.4.1 Basic Parameters**

##### *3.4.1.1 Magnification*

The magnification needed is calculated from the required resolution  $r$ , that is the size of the smallest detail in the object that we want to see, and the element size of the image sensor  $s$ . Then the magnification  $m$  is:

$$m = \frac{2s}{r} \tag{3.1}$$

The 'multiply by 2' is introduced to satisfy the needs of the sampling theorem. Applying this to the pollen imaging microscope and using a CCD with an element size of  $9\mu\text{m}$ , and a object needing to be resolved to  $0.9\mu\text{m}$  (as discussed earlier) requires an optical system that magnifies 20 times.

It is not possible to produce perfect images. Image fidelity is limited by lens aberrations and ultimately, diffraction. However if the degradation cannot be resolved by the sensor or perceived by a human then the image can for practical purposes be considered to be perfect.

### 3.4.2 Early Development

The original approach to the development of a computer microscope was an experimental one making use of existing cameras and lenses that were available. The focus of the research was to investigate problems highlighted by optical theory or to investigate reasons for conventional microscope design.

#### 3.4.2.1 Geometric Distortion

Geometric distortion (Section 2.2.1.3) is the easiest aberration to identify as it characteristically distorts squares, doing so with greatest severity away from the optical axis.

The test object used to determine distortion was the England Finder, a  $1\text{mm}^2$  grid, where each square is divided into 5 sections: a middle circle and four corner areas. The entire 3" slide is covered by these squares. Figure 9 shows one square (suitably magnified).

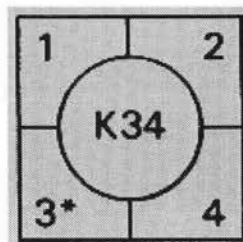
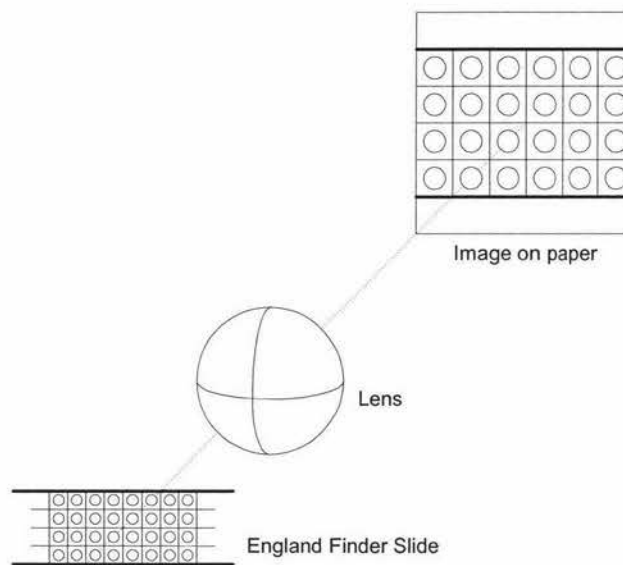


Figure 9 A square from an England Finder

To determine if the distortion was significant a piece of white paper was located vertically in place of the camera, as illustrated in Figure 10. Onto this a large image of the England Finder was projected. From this the distortion could be evaluated over a much larger area than the image sensor. If distortion were not present over this large area then it would be unlikely for it to be a problem in the small image sensor area.



**Figure 10 Distortion Test Setup**

Visually, the grid produced by the England Finder was square up to at least 30mm from the centre of the image sensor. This was determined by plotting some corner points from the image onto the paper and measuring the angles and dimensions of the lines that connected these points.

In conclusion, using a 50mm lens and 10x magnification, there is no significant distortion to the image. This is as expected because the magnitude of distortion increases as light rays from the object are increasingly off-axis [33] and in this case the rays are very close to the optical axis.

#### 3.4.2.2 Chromatic Aberration

Chromatic aberration (Section 2.2.1.3) is the most readily recognized optical aberration. It is familiar from looking through magnifying glasses. It is therefore an easy target for elimination.

The effect of chromatic aberration depends on the range of wavelengths in the light. If the range of wavelengths can be reduced, then the dispersion will be limited. To do this a narrow-band light source is required.

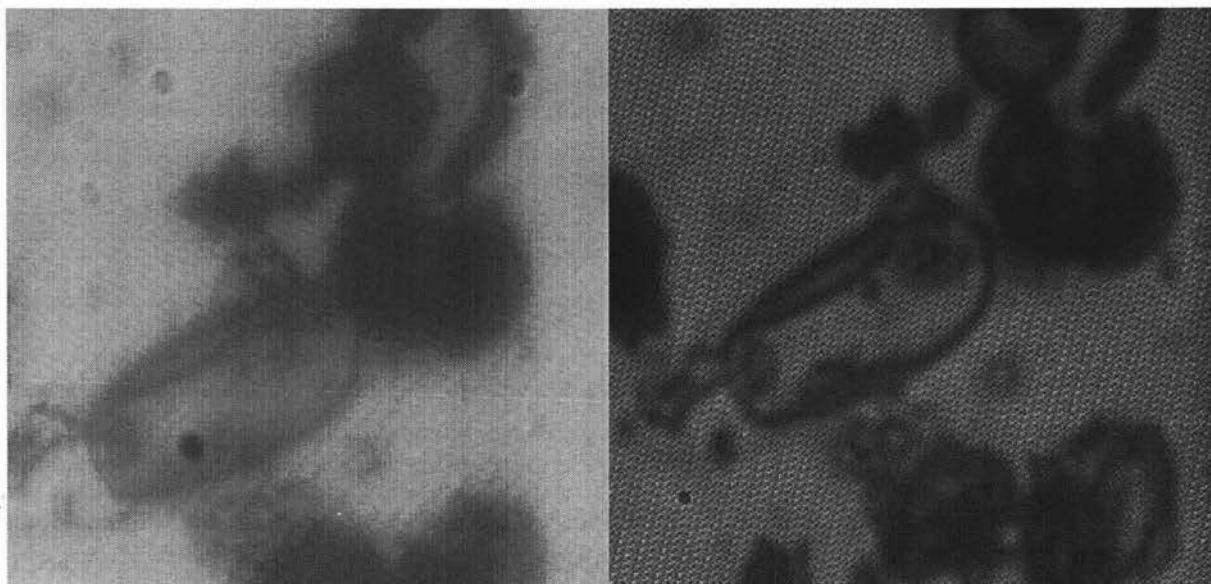
---

[33] M. Born, E. Wolf, "5.3 The primary (Seidel) aberrations" in *Principles of Optics*, 4<sup>th</sup> Ed., Pergamon Press. 1970. pp.211-218.

A means of creating narrow-band light is to filter another light source. However filtering reduces the intensity of the original light because it strips away the intensity provided by the unwanted wavelengths, so in order for it to be feasible a more intense light source or a longer integration time is required.

To provide adequate light intensity a 250W projector bulb was used. An RG-630 red 630nm high-pass (in wavelength) filter was placed in the light path. An IR-Stop filter rejecting wavelengths greater than 700nm was placed in front of the camera. This leaves a pass-band 70nm wide.

The left image of Figure 11 was taken without the filters in place. This image is slightly saturated because of the intensity of the light (despite using a 1/10000s shutter speed) and is blurred. The image on the right was taken with the filters in place and it can be seen that the objects have significantly better contrast, particularly the large ovaloid near the centre. Note the shutter speed has been reduced to 1/500s.



**Figure 11** The effect of applying a narrow-band filter

#### 3.4.2.3 Scatter

Light paths in a conventional microscope are shielded by matte black tubing that prevents light, other than that from around the specimen, being visible at the eyepiece. This reduces scatter, that is spurious light, from interfering with the image. Scatter was removed from the microscope design by placing matte black paper inside the optical tubes (made of white down-pipe) and shielding the apparatus from stray light with a blackout cloth.

#### 3.4.2.4 General Problems

Generally these early experiments did not achieve clear images. While the reason for this was not identified at this stage, several contributing elements were:

- The lighting was not condensed as it is in a conventional microscope
- The camera introduced noise from a colour sub-carrier as can be seen in the right of Figure 11.
- The equipment was not sufficiently precise to allow fine adjustments of focus and equipment position.

To eliminate the latter two problems an optical bench and equipment was purchased from Edmund Optics [34], a Micropix M-1024 monochrome Firewire digital camera was purchased from Turnkey Solutions [35], and custom fittings were designed and manufactured by JJ Niven Engineering [36]. This required a recalculation of the basic parameters. The Micropix M-1024 has square elements with a dimension of  $4.65\mu\text{m}$ . Using the formula from section 3.4.1.1 with  $s=4.65\mu\text{m}$  and  $r=0.9\mu\text{m}$  the magnification required is equal to  $10.3\times$ . Generally it is preferable to over-sample if the sampler has adequate storage, as the  $1024\times 768$  pixel CCD in the M-1024 has. Therefore a magnification of  $12\times$  to  $16\times$  would be used in practice.

---

[34] Edmund Industrial Optics, 101 East Gloucester Pike, Barrington, New Jersey, +1 800 363-1992, [www.edmundoptics.com](http://www.edmundoptics.com)

[35] Total Turnkey Solutions, Mona Vale, NSW, Australia, +61 2 9979 5643, [www.turnkey-solutions.com.au](http://www.turnkey-solutions.com.au)

[36] JJ Niven Engineering, 38 Armstrong St., Palmerston North. +64 6 3574039.



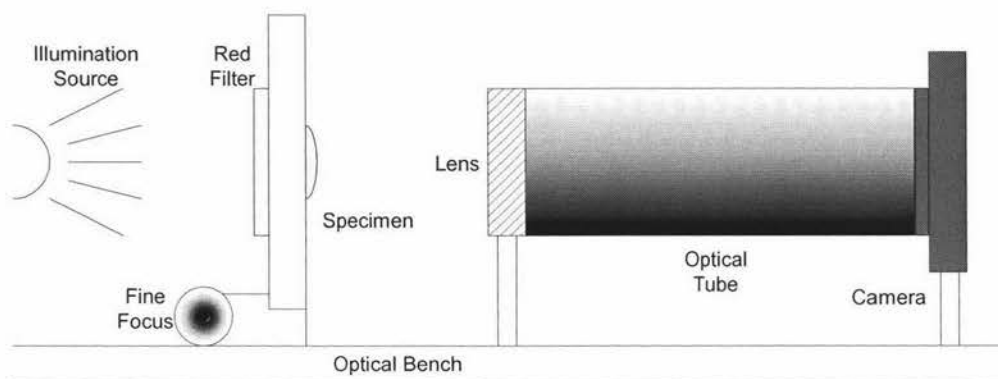
### 3.4.3 Experiments to Determine Resolution

Earlier experiments were somewhat ad hoc and it was difficult to determine whether the optical system set-up or the poor equipment was causing the image degradation. By acquiring precise optics and equipment, it was determined that poor equipment was not the primary fault causing unclear images. Therefore further experiments were needed to be conducted into the cause of the problems.

The aim of the first experiment was to:

- Determine that the basic parameters calculated in theory were equal, within tolerances, to those found in practice; and
- Determine if the degradation in quality is due to this microscope.

In these experiments the distance was varied between the optical components used to image a test slide of a grid made of  $10\mu\text{m}$  squares. The experimental setup is shown in Figure 12. For each experimental setup the largest rectangle (on the image of the grid) is used to calculate the measured parameters of the system.



**Figure 12 Experimental Setup**

There are some measurement errors involved that are estimated. They come from two places:

- Grid Rotation. Faulty alignment in the test equipment resulted in the largest rectangle on the image of the grid to be rotated slightly relative to the image sensor.
- Line width. The grid lines have considerable width (probably in the order of  $2\text{-}3\mu\text{m}$ ), which is not uniform. Therefore the middle of the line is estimated.



The errors are calculated by finding the corner of the largest rectangle that is furthest away from a corner of the grid, and counting the number of pixels (in both dimensions) between that grid corner and the corner of the rectangle. The estimated measurement error is specified in the results of the experiment.

The magnification values used provide a range that allowed comparison with previous images acquired using the setup and camera used to take the images in Figure 11 captured at 20× magnification. The magnification needed theoretically is between 10× and 16× and therefore the range of magnifications includes all of these values.

Note the tolerances of the largest rectangle size are not known, as the manufacturer of the grid did not provide a specific error measure. The width of the grid is nominally 1µm, with an unspecified tolerance.

Pixel Dimension = Dimension Size (µm) / Dimension Pixel Count

Actual Magnification = 4.65(µm) / Pixel Dimension (µm) where 4.65µm is the size of the sensing elements.

#### 3.4.3.1 Experimental Results

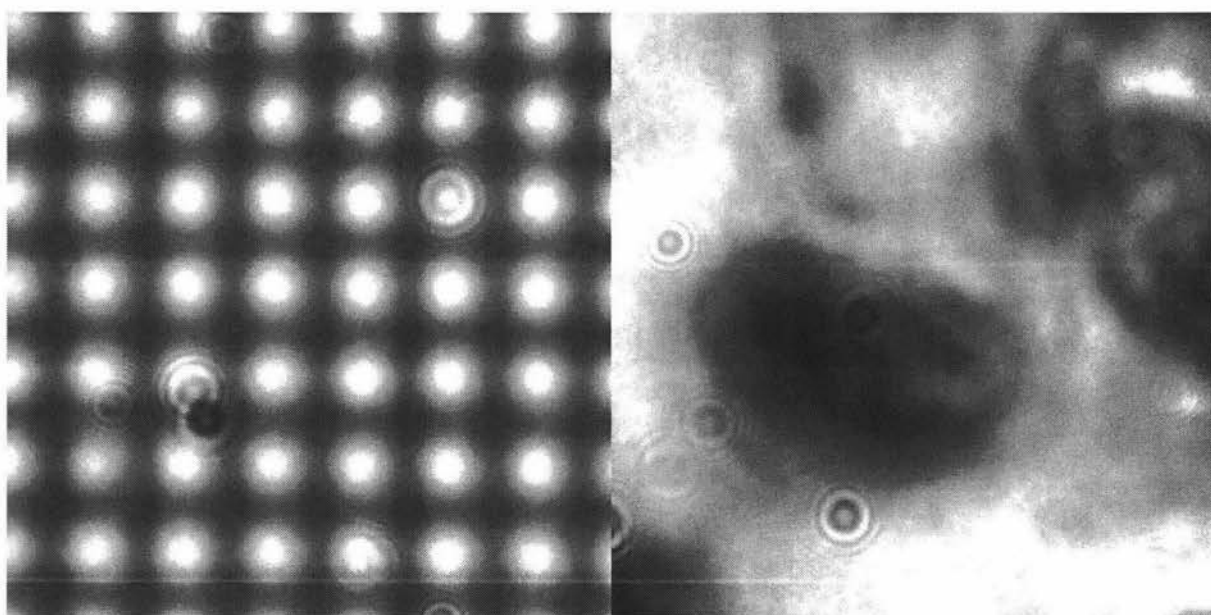
##### 20x

*Theoretical Values:*

S <sub>i</sub>	530mm	f	25mm	M	20.2x
----------------	-------	---	------	---	-------

*Measured Results:*

Largest Rectangle	230 × 160 µm	tolerances unknown.
	1001 × 713 pixels	± 6 × 14 pixels
Pixel Width	0.23 µm	± 0.5%
Pixel Height	0.22 µm	± 2%
Actual Magnification: Width	20.2	± 0.5%
Actual Magnification: Height	20.7	± 2%



**Figure 13 Sections (300 × 300 pixels) of Images Captured at 20x Magnification**

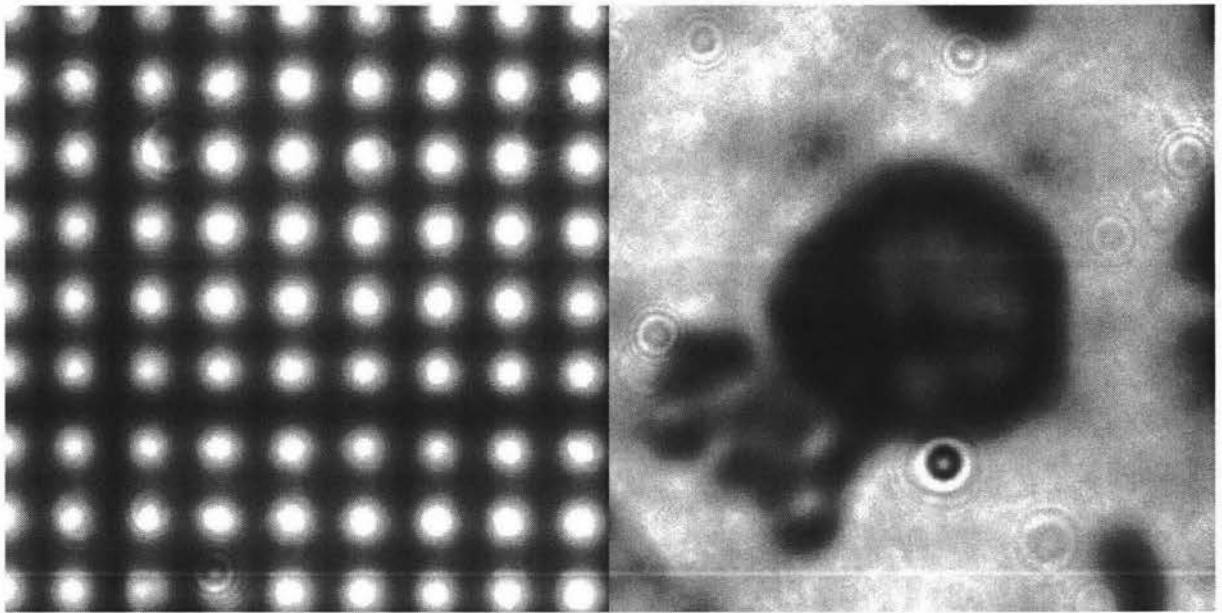
**16x**

*Theoretical Values:*

$S_i$	430mm	$f$	25mm	$M$	16.2x
-------	-------	-----	------	-----	-------

*Measured Results:*

Largest Rectangle	280 × 200 $\mu\text{m}$	tolerances unknown.
	999 × 727 pixels	$\pm 8 \times 10$ pixels
Pixel Width	0.28 $\mu\text{m}$	$\pm 1\%$
Pixel Height	0.28 $\mu\text{m}$	$\pm 0.8\%$
Actual Magnification: Width	16.6	$\pm 1\%$
Actual Magnification: Height	16.6	$\pm 0.8\%$



**Figure 14 Sections (300 × 300 pixels) of Images Captured at 16x Magnification**

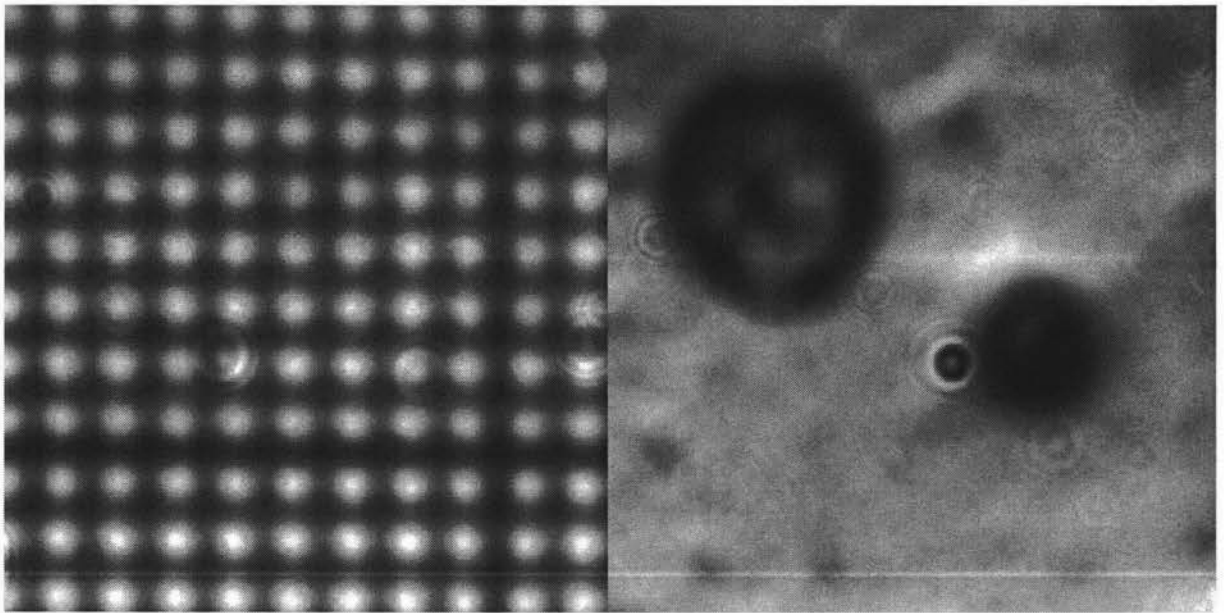
**13x**

*Theoretical Values:*

$S_i$	350mm	$f$	25mm	$M$	13x
-------	-------	-----	------	-----	-----

*Measured Results:*

Largest Rectangle	$340 \times 250 \mu\text{m}$	tolerances unknown.
	$976 \times 734 \text{ pixels}$	$\pm 7 \times 11 \text{ pixels}$
Pixel Width	$0.35 \mu\text{m}$	$\pm 0.7\%$
Pixel Height	$0.34 \mu\text{m}$	$\pm 1.5\%$
Actual Magnification: Width	13.3	$\pm 0.7\%$
Actual Magnification: Height	13.7	$\pm 1.5\%$



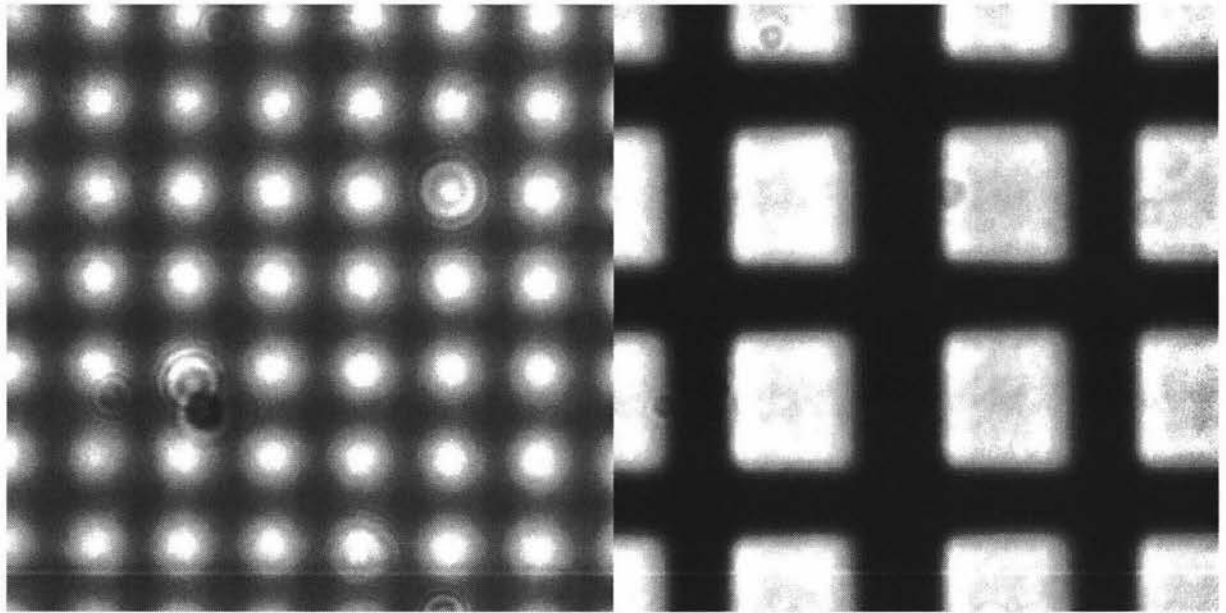
**Figure 15 Sections (300 × 300 pixels) of Images Captured at 13x Magnification**

#### *3.4.3.2 Errors*

The results should give the same magnification in both dimensions. For 20x and 13x this is not the case, however the pixel width and height in both cases varies by 0.1 $\mu$ m. As this is also the smallest value we use there is inherent error in rounding to this level of significant figure. Despite these errors, the values for width and height are very similar.

#### *3.4.3.3 Comparison to conventional compound microscope*

In order to determine if the degradation in quality is due to this microscope, it is easiest to compare images taken using this microscope with those from a high quality conventional compound microscope. Figure 16 shows the difference in quality between this microscope and a compound microscope. It is valid to use different magnification in this case as it would be expected that a high magnification would give a poorer image due to empty magnification. So there is clearly a problem given that image captured by this system at 20x is significantly worse than that captured at 40x using a compound microscope



**Figure 16 Microscope Contrast Comparison. Left: 20x Setup. Right: Compound Microscope 40x**

#### *3.4.3.4 Conclusions*

The basic parameters of the system calculated in theory were equal, within tolerances, to those found in practice. Secondly the images generated by this system are significantly worse than those from a compound microscope. Therefore there are further factors causing image degradation.

Three factors that could introduce the image degradation are:

- Blurring Aberrations such as spherical aberration and coma
- Diffraction effects
- Inadequate lighting as the experimental system does not have a condenser or illumination train

### **3.4.4 Identifying and Quantifying the sources of Image Degradation**

#### *3.4.4.1 Aberrations*

To study the effect of blurring aberrations requires the analysis of the rays travelling through the lens. This is a mathematically difficult operation for all cases, but more so for the off-axis

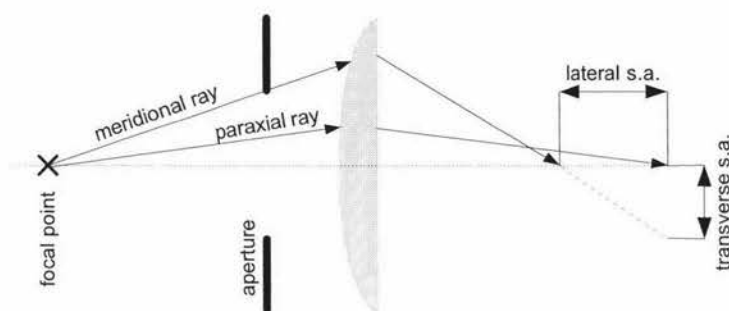
aberrations coma and astigmatism. Therefore spherical aberration will be analysed first, and only if it proves not to be a factor will the other blurring aberrations be investigated.

There are two key factors that affect the spherical aberration:

1. The aperture size limits the maximum ray divergence from the object, limiting the divergence of meridional rays. Therefore spherical aberration is reduced as the aperture is closed.
2. The refractive index of the lens determines the extent of the refraction at the lens surfaces. A smaller refractive index causes less divergence and therefore less spherical aberration.

The modelling was performed in Excel™ to allow the results of manipulating input variables to be studied quickly. The ray-tracing method used is taken from [37].

To calculate Spherical Aberration (s.a.) the two most distant points of focus are required. These are generated by rays which are very close to the optical axis (paraxial), and rays which originate from the optical axis, but diverge to the edge of the aperture, that is are the rays with the greatest angle to the optical axis, called meridional rays. The distance between the points at which these two rays converge to the optical axis (on the image side) is called the lateral spherical aberration. The consequential degradation in the image is how widely they are spread onto the image plane, called the transverse spherical aberration. Figure 17 illustrates this concept.



**Figure 17 Spherical Aberration (s.a.)**

It is important to note that spherical aberration does not result in a uniform blur, but the effect can be considered as a modification to the point spread function. In the case of spherical aberration, there is a simple radial blurring.

---

[37] Warren J. Smith, *Modern Optical Engineering*, McGraw-Hill, 1966, Chapters 3 and 10, pp.49-71 and 247-279

### *Model Parameters*

Lens focal length	25mm
Distance from focal point to aperture	14mm
Distance from aperture to lens	11mm
Aperture Radius	0.4mm

### *Notes:*

- The refractive indices used are those of available lenses and have been taken from the relevant glass tables assuming a wavelength of  $\lambda=665\text{nm}$ .
- The ray trace begins with rays emanating from the focal point of the lens on the object side.

### *Results*

Magnification	Refractive Index	Transverse s.a. ( $\mu\text{m}$ )
20	1.665	30.5
16	1.665	23.7
12	1.665	17.1
12	1.514	2.5

These results show that spherical aberration decreases with magnification, as is expected. What these results do not show is that changing the aperture size made no difference to the image quality. Furthermore, reducing the refractive index of the lens (by using a different glass) should have improved the image quality significantly as the results show a reduction in transverse spherical aberration by an order of magnitude. However no significant improvement in the image quality was seen.

This leads to the conclusion that spherical aberration is not the primary cause of image degradation in this optical system. This is not a simple conclusion however. Aberration cannot be treated in isolation, and it's possible that non-parallel lighting is creating rays with greater



angles than we would expect. In other words there could be a confounding relationship between aberrations and illumination.

#### 3.4.4.2 Diffraction

The second potential cause of degradation is diffraction. It can be shown for far-field diffraction that the spread of the central maximum of the diffraction pattern emanating from a point within a slit (a 1-D simplification of an aperture) has the following formula:

$$W = \frac{2L\lambda}{b} \quad (1.2)$$

Where  $W$  is the maxima width,  $L$  is the distance from the slit to the image,  $b$  is the slit width, and  $\lambda$  is the wavelength of the light [38].

Application of this equation with the values from the 20x optical setup with maximum aperture and standard aperture is displayed below:

Maximum Aperture:  $b = 21\text{mm}$ ,  $L = 525\text{mm}$ ,  $\lambda = 700\text{nm}$  gives  $W = 35\mu\text{m}$

Standard Aperture:  $b = 4\text{mm}$ ,  $L = 525\text{mm}$ ,  $\lambda = 700\text{nm}$  gives  $W = 184\mu\text{m}$

Both of these values immediately stand out as being much larger than the intended resolution of  $2\mu\text{m}$ . Diffraction is clearly a major factor in the blurring of the images.

#### 3.4.4.3 Illumination

Conventional compound microscopes have an illumination train that provides Köhler illumination (section 2.3.2.2). This provides parallel and uniform illumination of the specimen. The experimental optical system does not provide this quality of illumination. However as the illumination is theoretically not critical at low optical magnifications and as it has been shown that spherical aberration and diffraction are a significant problem, the illumination will not be addressed unless it is shown that it is the only remaining fault. The reason for this is that throughout this analysis the aim has been to design the simplest possible microscope, a single lens, and adding optics to provide correct illumination requires adding lenses.

---

[38] F. L. Pedrotti, L. S. Pedrotti, "16-2 Beam Spreading" in *Introduction to Optics*, 2nd Ed. New Jersey: Prentice-Hall, 1993, pp329-330.



## 3.5 Conclusion

### 3.5.1 Feasibility of a Single Lens Microscope

Correcting for spherical aberration requires that the aperture diameter be decreased. However it can be seen from equation (1.2) that decreasing the aperture (parameter  $b$ ) increases diffraction. Thus we have a trade-off between aberration and diffraction.

Applying equation (1.2) shows that an aperture greater than 21mm is required to reduce the effect of diffraction to the desired value of  $2\mu\text{m}$ . However this increase in aperture size increases the quantity of non-paraxial rays in the system and increases the magnitude of the aberrations.

In order to correct for spherical aberration the diameter of the point spread function should be at most  $2.2\mu\text{m}$ . To achieve this, an aperture, placed in the same axial position as in the model in section 3.4.4.1, with a diameter 0.061mm is required. At this aperture diameter the effect of diffraction would be a point spread function with diameter 12mm, which is greater than the area of the imaging sensor! Additionally an aperture of this size would block out 99.9% of the image.

This trade-off in the optical system prevents achievement of the optimal resolution. Although the values here are for the case of a lens with  $f=25\text{mm}$ , a lens where the diameter of both the spherical aberration and beam spread is smaller than  $2\mu\text{m}$  cannot be found. This is despite applying ray tracing and beam spread equations to optical systems in which the lens has both shorter and longer focal lengths than 25mm. Therefore, a single lens microscope magnifying twenty times cannot provide the resolution required for imaging pollen. This explains the blurring of the images at  $20\times$  magnification, and most likely for those at lesser magnification.

### 3.5.2 Alternatives

Given that a single lens microscope at a magnification of  $20\times$  or more is not possible there are two alternatives to consider for the basic design of a computer microscope.

The first option, custom optics, allows design flexibility. Triplets, such as Cooke's Triplet, composed of two positive outer lenses and a negative inner lens, has just enough degrees of

freedom to allow the designer to correct all the primary aberrations [39]. However considerable optical and mechanical design is required.

The second option, a standard finite achromatic microscope objective, is less flexible than custom optics as the design does not specifically suit the optical problem it is being used to solve. However a standard objective has known design parameters, is mechanically housed, and contains corrective optics for all primary aberrations. Importantly, the cost of a standard objective and the cost of three lenses plus housing are approximately equal [40].

As this thesis is essentially an exercise in system integration the standard microscope objective is the prudent and expedient option and has been selected for the design.

---

[39] W. J. Smith, "The Design of Optical Systems" in *Modern Optical Engineering*, USA: McGraw-Hill, 1966, pp.340-347

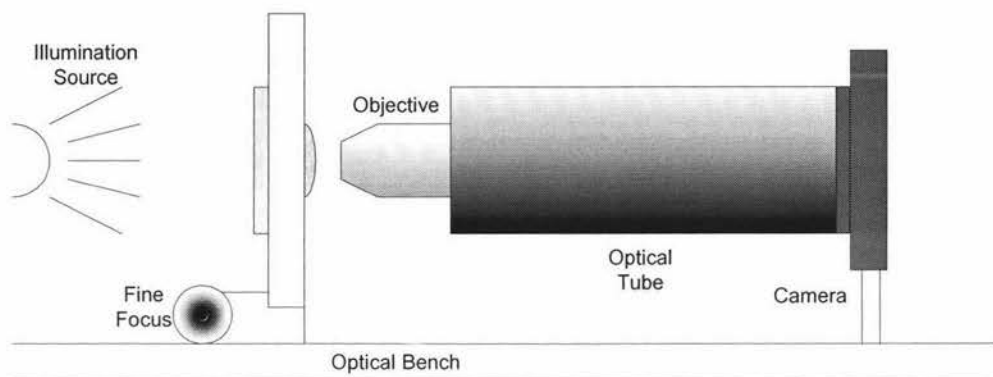
[40] Edmund Optics, *Optics and Optical Instruments Catalog*, New Jersey. 2002. Online copy available at [www.edmundoptics.com](http://www.edmundoptics.com)

## 4 Image Capture System Design

The microscope now consists of:

- A CCD camera
- A tube which screws into the camera using a standard lens mount thread ('C' or 'F' mount)
- A standard microscope objective

Figure 18 illustrates the microscope within the experimental platform.



**Figure 18 Experimental Setup**

Having selected the general design, this chapter examines the components required for a computer microscope and for viewing pollen.

### 4.1 Selecting an Objective

Standard microscope objectives come in magnification of  $10\times$  and  $40\times$ , with a standard tube length (distance from the objective to the image plane) of 160mm. Some catalogs also list  $20\times$  objectives that would be most suitable to this application. However as access was immediately available to only  $10\times$  and  $40\times$  objectives, these were used to capture test images.

The aim of the following experiments is to examine the image quality and magnification of certain arrangements of objective and camera. These experiments will help to determine the most appropriate setup to use in the final design.

### Standard 10x Objective

*Theoretical Values:*

M	10x	Tube Length	180mm $\pm$ 10mm
---	-----	-------------	------------------

*Measured Results:*

Largest Rectangle		390 $\times$ 290 $\mu$ m	tolerances unknown.
		1000 $\times$ 738 pixels	$\pm$ 7 $\times$ 15 pixels
Pixel Width		0.39 $\mu$ m	$\pm$ 0.7%
Pixel Height		0.39 $\mu$ m	$\pm$ 2%
Actual Width	Magnification:	11.9	$\pm$ 0.7%
Actual Height	Magnification:	11.9	$\pm$ 2%

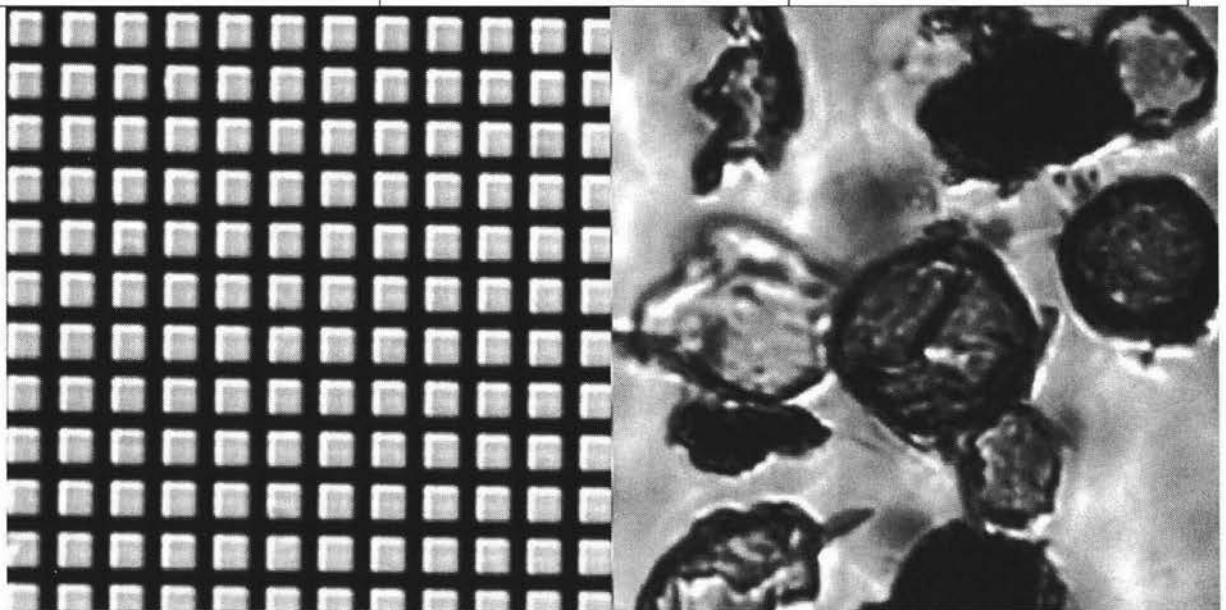


Figure 19 Sections(300  $\times$  300 pixels) of images captured with a 10x Objective setup at a magnification of 12x

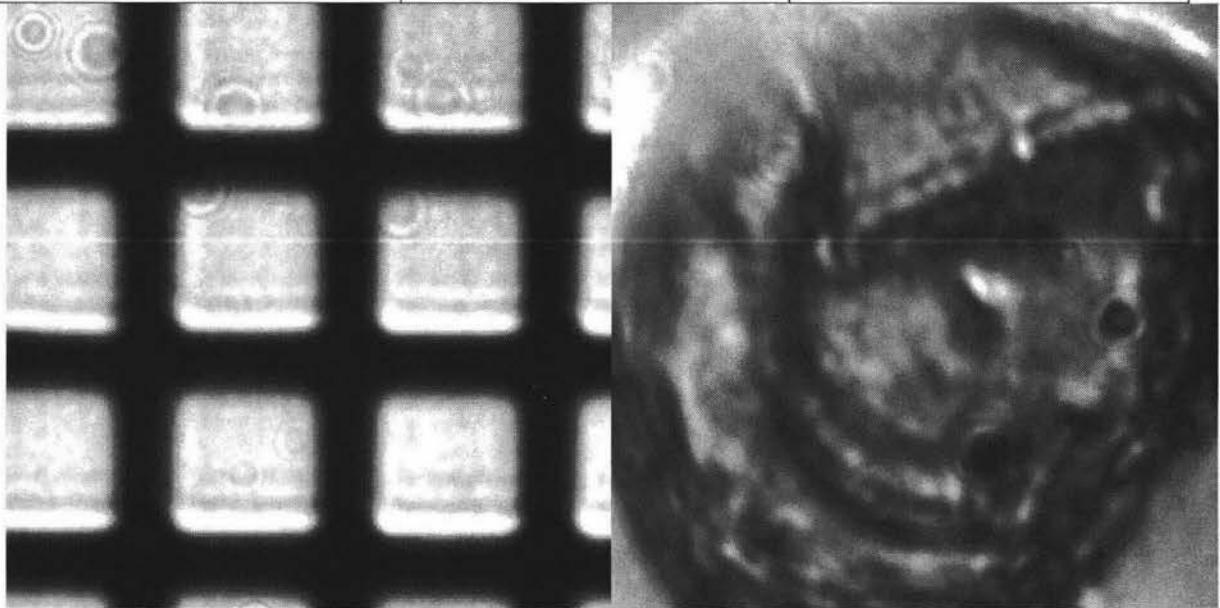
### Standard 40x Objective

*Theoretical Values:*

M	40x	Tube Length	180mm $\pm$ 10mm
---	-----	-------------	------------------

*Measured Results:*

Largest Rectangle	90 × 60 μm	tolerances unknown.
	898 × 611 pixels	± 8 × 14 pixels
Pixel Width	0.10 μm	± 0.9%
Pixel Height	0.098 μm	± 2%
Actual Magnification: Width	46.5	± 0.9%
Actual Magnification: Height	46.5	± 2%



**Figure 20** Sections(300 × 300 pixels) of images captured with a 40x Objective setup at a magnification of 46.5x

### Discussion

From the above it can be seen that the 46.5× images are quite blurred due to empty magnification from selecting a magnification whose theoretical resolution would exceed the limits of optical resolution. The 12× images are almost as large as are required for classifying pollen and if the optical tube length were increased the necessary magnification may be obtained. It is necessary to confirm that doing this does not degrade the image quality.

### 10x Objective to Magnify 15x

*Theoretical Values:*

M	unknown	Tube Length	220mm ± 10mm
---	---------	-------------	--------------

*Measured Results:*

Largest Rectangle	$300 \times 230 \mu\text{m}$	tolerances unknown.
	$975 \times 745 \text{ pixels}$	$\pm 10 \times 16 \text{ pixels}$
Pixel Width	$0.31 \mu\text{m}$	$\pm 1\%$
Pixel Height	$0.31 \mu\text{m}$	$\pm 2\%$
Actual Magnification: Width	15	$\pm 1\%$
Actual Magnification: Height	15	$\pm 2\%$

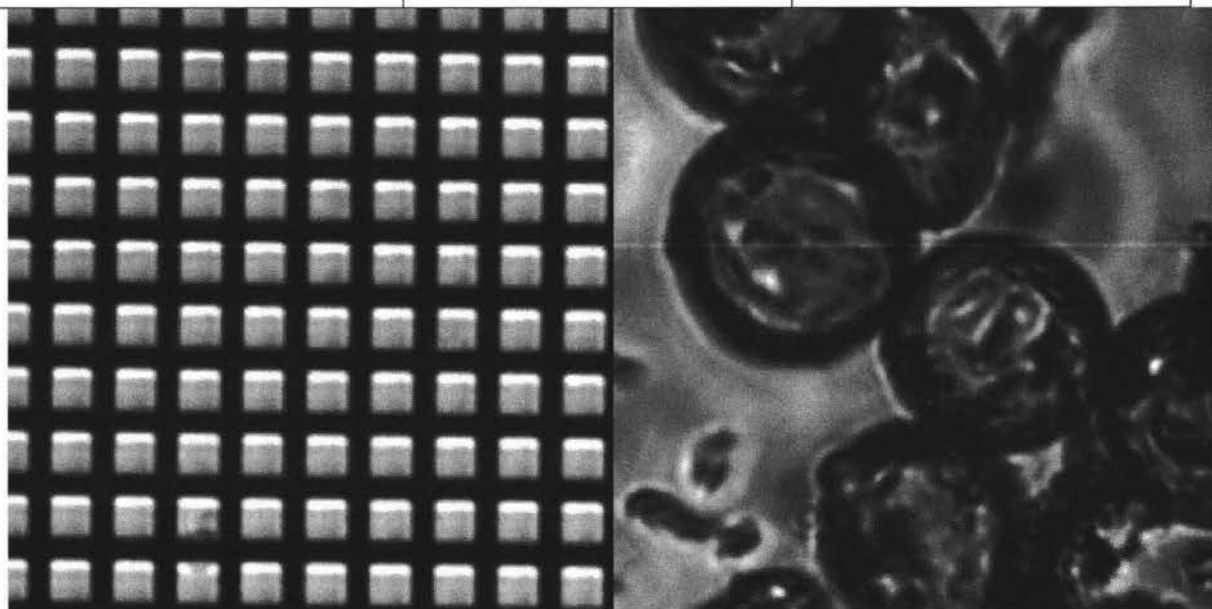


Figure 21 Sections( $300 \times 300 \text{ pixels}$ ) of images captured with a 10x Objective setup at a magnification of 15x

### 10x Objective to Magnify 22x

*Theoretical Values:*

M	unknown	Tube Length	$330\text{mm} \pm 10\text{mm}$
---	---------	-------------	--------------------------------

*Measured Results:*

Largest Rectangle	$210 \times 160 \mu\text{m}$	tolerances unknown.
	$954 \times 746 \text{ pixels}$	$\pm 8 \times 14 \text{ pixels}$
Pixel Width	$0.22 \mu\text{m}$	$\pm 0.8\%$
Pixel Height	$0.21 \mu\text{m}$	$\pm 2\%$
Actual Magnification: Width	21	$\pm 0.8\%$
Actual Magnification: Height	22	$\pm 2\%$

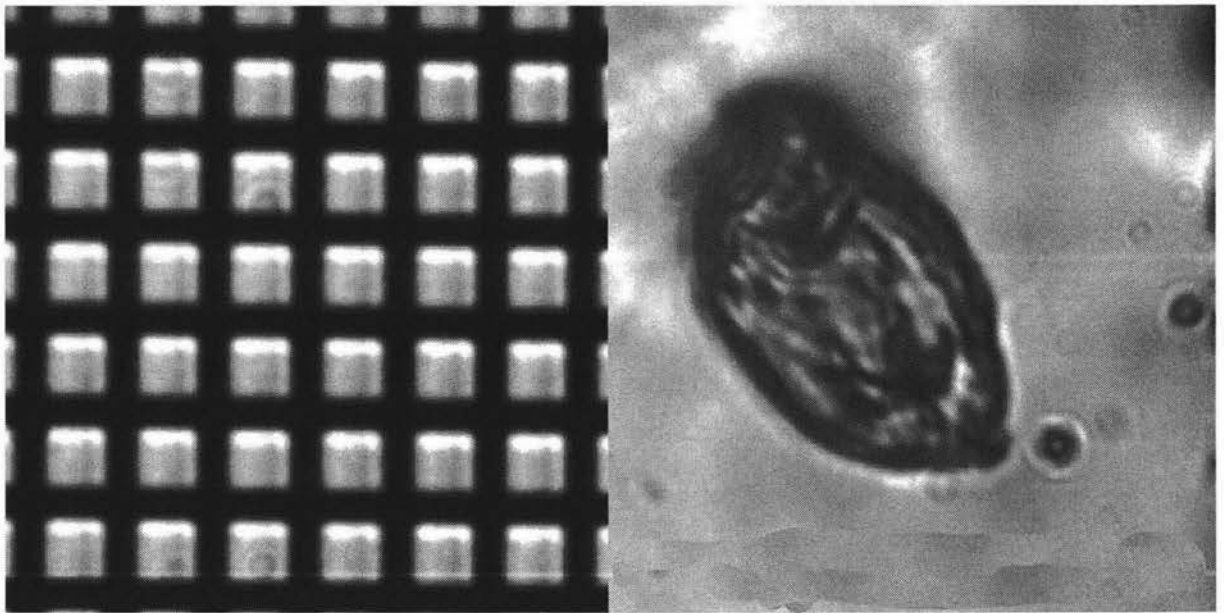


Figure 22 Sections(300 × 300 pixels) of images captured with a 10x Objective setup at a magnification of 22x

#### 40x Objective to Magnify 27x

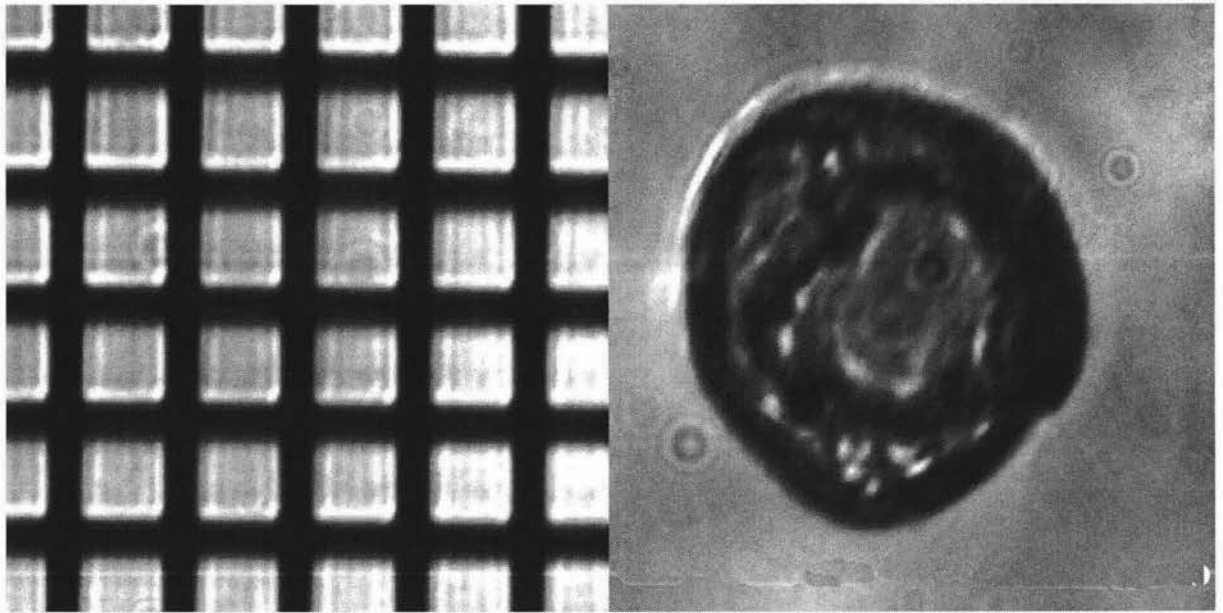
*Theoretical Values:*

M	unknown	Tube Length	90mm ± 10mm
---	---------	-------------	-------------

*Measured Results:*

Largest Rectangle	170 × 120 μm	tolerances unknown.
	975 × 706 pixels	± 9 × 13 pixels
Pixel Width	0.17 μm	± 0.9%
Pixel Height	0.17 μm	± 2%
Actual Magnification: Width	27	± 0.9%
Actual Magnification: Height	27	± 2%





**Figure 23** Sections(300 × 300 pixels) of images captured with a 40x Objective setup at a magnification of 27x

#### *4.1.1.1 Discussion*

At 22x magnification, the highest magnification obtained using the 10x Objective, the quality of the image is degraded from what is expected. Similarly, using the 40x Objective to optically magnify 27x produces out of focus images. As the image captured at close to the standard 40x magnification in Figure 20 is also out of focus, this leads to the conclusion that increasing the optical magnification does not improve image quality.

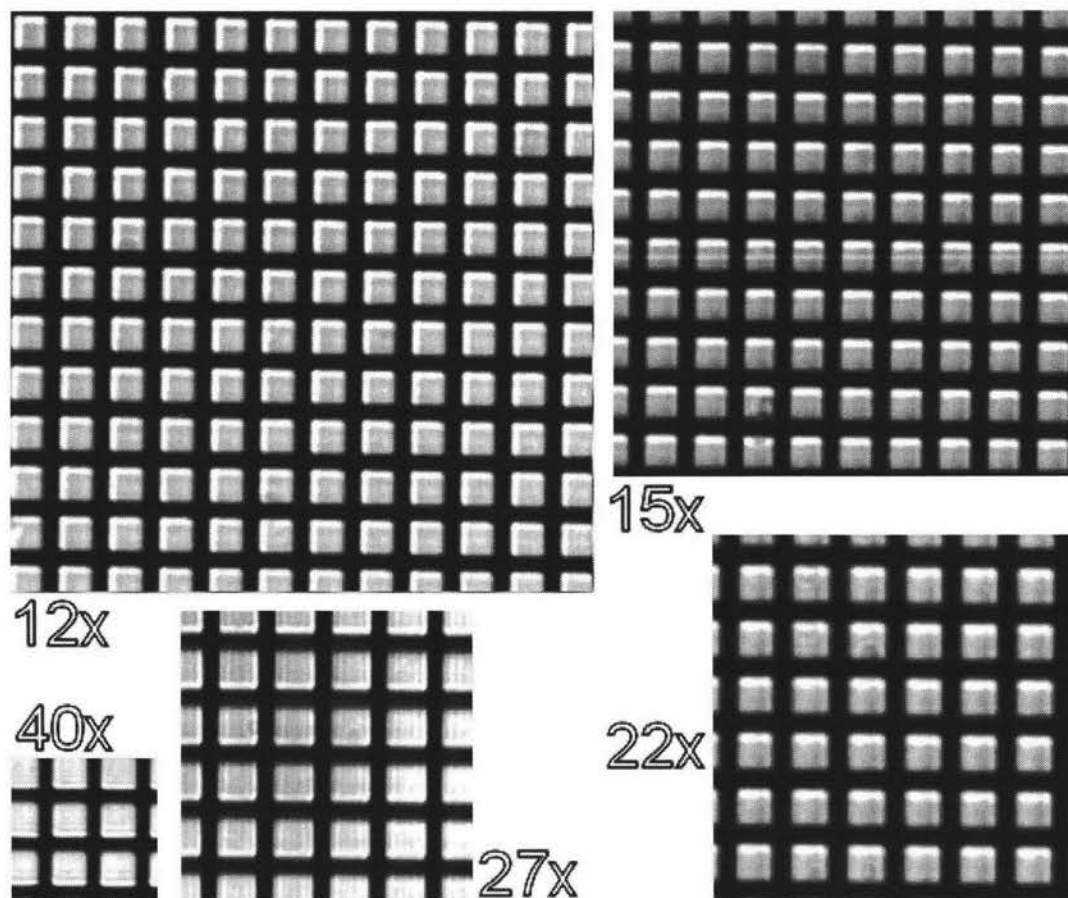
However from these images it cannot be directly determined if exceeding the design parameters of the microscope objectives is also causing image degradation. If the degradation is from empty magnification alone, then reducing the scale of the images in proportion to its magnification should produce the same results. Any significant differences in the rescaled images would be from the effects of the objective. Minor differences may arise from resampling and general capture inconsistency.

Figure 24 shows the results of this. The only image to show significant degradation is the image captured at 22x magnification. The degradation in this image could be either from poor focus or exceeding the design limitations of the objective. However given that the quality of the



remaining images is consistent then it seems reasonable to put the poor quality of the image captured at 22x magnification down to poor image focus during capture.

In conclusion, it is acceptable to use the standard microscope objectives at different optical tube lengths from 160mm. From these experiments, the acceptable optical tube length of the 10x objective is at least the range 160-330mm, and the 40x objective 90-180mm.



**Figure 24 Comparative Image Quality of images at different magnifications. All the images are scaled to 12x magnification.**

#### *4.1.1.2 Conclusion*

Based on the quality of the images presented in the preceding 6 figures, the 12x magnification is the most appropriate magnification to use, as the images obtained using that setup contain crisp edges and high contrast. As the magnification increases beyond this the images became progressively more blurred and reduced in contrast due to empty magnification.

## 4.2 Camera Requirements

A key parameter of the design is the element size of the image sensor. This value was factored into the design at earlier stages of the development (section 3.4). A smaller element size proportionally reduces the optical magnification required. However if the optical magnification is increased then effects associated purely with optical magnification could become a problem, such as the need for critical illumination.

To test the effect of increasing the element size a second camera, with element size  $9.0\mu\text{m}$ , has been introduced to the system. The increased element size of this camera requires an increase in the optical magnification to achieve the same system magnification. This will require an increased optical tube length. The aim of this experiment is to determine if there are differences in the quality of the images at the same magnification.

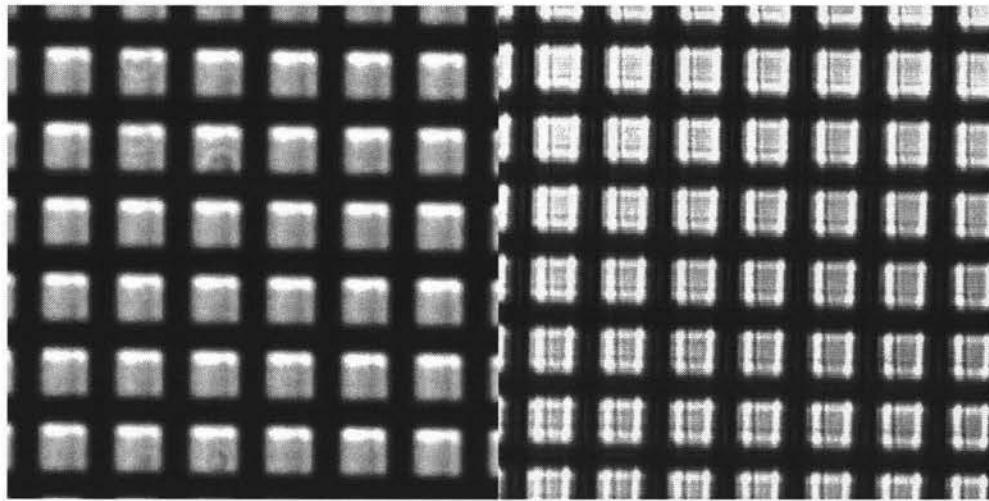
### Standard 40x Objective – Increased Camera Element Size

*Theoretical Values:*

M	40x	Tube Length	$140\text{mm} \pm 10\text{mm}$
---	-----	-------------	--------------------------------

*Measured Results:*

Largest Rectangle	$220 \times 170 \mu\text{m}$	Tolerances unknown.
	$1000 \times 800 \text{ pixels}$	$\pm 17 \times 22 \text{ pixels}$
Pixel Width	$0.22 \mu\text{m}$	$\pm 2\%$
Pixel Height	$0.21 \mu\text{m}$	$\pm 3\%$
Optical Magnification: Width	41	$\pm 2\%$
Optical Magnification: Height	43	$\pm 3\%$



**Figure 25** Image captured at different optical magnifications having the same system magnification.

#### *4.2.1.1 Discussion*

A 40x objective was necessary to generate an image with a suitable size for imaging pollen, as set in section 4.1. The images shown in Figure 25 compare those taken with a camera of element size  $4.65\mu\text{m}$  and an optical magnification of 22x (Figure 25 left) with those taken with a camera of element size  $9.0\mu\text{m}$  and an optical magnification of 42x (Figure 25 right). The image of the left is superior to that on the right, however at such a high optical magnification the image is very difficult to focus. More importantly, at this optical magnification the lighting becomes critical which may also account for the degradation of the image on the right. Regardless of the cause of the degradation, including correct illumination for a high powered objective in the system would require additional optics, which detracts from the aim of achieving simplicity in the design.

In conclusion, a small element size in the image sensor is important, as it reduces the optical magnification, which removes the criticality of the illumination, and makes the microscope easier to operate.

## 4.3 Selecting Illumination

### 4.3.1 Optical Wavelength

The effect of chromatic dispersion on the quality of an image has been discussed in section 3.4.2.2. This led to the addition of a narrow-band optical filter to the system. One question that remains from that section is why the narrow-band filter was placed at the red end of the spectrum. After all if the aim to see small objects, it would seem sensible to use the smallest wavelengths of light possible.

The choice is affected by three factors:

- The spectral response of the CCD
- The wavelength filters available
- The quality of the correction by the objective

The spectral response of the CCD in the Micropix M-1024 is shown in Figure 26. This would suggest using a filter centred around 500nm where the sensitivity of the CCD is greatest. However from the equipment available two potential filters are possible: the first 500nm – 600nm pass band, but with high transmittance (>20%) in the stop bands; or 630nm – 700nm pass band with less than 4% transmittance in the stop bands. The response of the CCD is still reasonable at 700nm, and the second filter has a smaller bandwidth than the first. Therefore the second filter was selected.

As noted above, the third factor is the quality of the correction by the objective. The 630nm - 700nm filtered image was compared with an image captured with only the 700nm IR-cut filter in place. The difference in image quality was barely perceptible. Therefore a narrow-band filter is not essential in this computer microscope design.

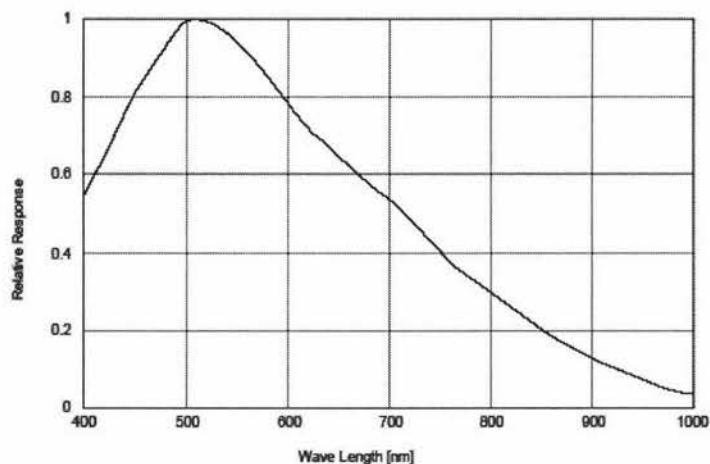


Figure 26 Spectral Response of the CCD in the M1024 camera

### 4.3.2 Illumination Source

A filtered light source requires a high intensity and high power white light source, and then removes most of the light intensity because it is not a useful wavelength. The alternative to doing this is to find suitable illumination that is already of the correct wavelength. Such sources are considered narrow-band light sources and come in three practical types: LEDs, arc lamps and lasers.

#### 4.3.2.1 LEDs

A light emitting diode (LED) is a semiconductor junction that emits light and heat when sufficient current is passed through it. The LEDs used to test the effectiveness of this form of illumination had a bandwidth of  $660 \pm 20$  nm [41]. Unlike incandescent sources, LEDs are directional, concentrating most of the light energy within  $10^\circ$  of the centre axis.

Nine LEDs arranged in a 3x3 grid arrangement were used to illuminate the specimen. This number of LEDs was used as their physical size made placing more LEDs impractical within the area necessary for illuminating the specimen. The light path was surrounded by a piece of white plastic down-pipe to ensure stray white light did not mix with the narrow-band light, and that divergent light from the LED's was reflected back into the tube.

---

[41] Dick Smith Electronics Catalogue Number Z4074. Light output: 3.6mcd at 20mA

The pattern of the LEDs was evident at the camera. Rather than a single 'mass' of light, there were 9 separate points. This is not useful as in effect only the centre LED is providing illumination. To merge the separate points a condenser lens, possible with one or more ancillary lenses, is required.

#### *4.3.2.2 Lasers*

A laser is a coherent light source producing a very intense and very narrow band of light. Due to mass production lasers are also very economical as a packaged laser diode retails for around \$2, with the added advantage of having an appropriate wavelength for a CCD sensor. Lasers present one problem in this application: coherence. The coherent nature of the laser light causes it to form interference patterns when it travels around objects, resulting in an image containing high contrast artefacts. It may be possible to deconvolve the image, but that is of far greater complexity than is practical. An attempt was made to remove the coherence by applying a prepared transparent surface etched with a fine abrasive in an endeavour to provide path differences of the order of  $\lambda/4$ . This quick and crude experiment was not successful.

#### *4.3.2.3 Arc Lamps*

The scientific mercury arc lamp that I was able to obtain was of insufficient intensity to illuminate the source adequately, both due to its lack of intensity and the lack of a dark housing that would normally be found with an arc lamp in a conventional microscope. Furthermore, an arc lamp operates at high voltage requiring a bulky step-up transformer or the switch-mode equivalent, both of which limit the flexibility of the design. However the key drawback of arc lamps, particularly when compared to LEDs and lasers, is its expense, prohibiting its use in this design.

#### *4.3.2.4 Conclusion*

LEDs or a laser provide the most economic solution. LEDs have the advantage of being the most flexible, in that they can be flashed, or 'ganged up' relatively easily. Lasers have the advantage of being the most intense. However both of these options require extra lenses to either form the light from the LEDs into a single beam, or to disperse and remove the coherence of the laser. Either solution would therefore detract from the aim of design simplicity.

It has been found that a standard 75W incandescent light bulb provides sufficient light intensity and an adequately uniform field to be used as a light source, even with the 70nm-wide filter in place. The light bulb was placed 25cm behind the specimen slide in the experimental setup. This is the illumination scheme used in the final design.

#### **4.4 Mechanical Discussion**

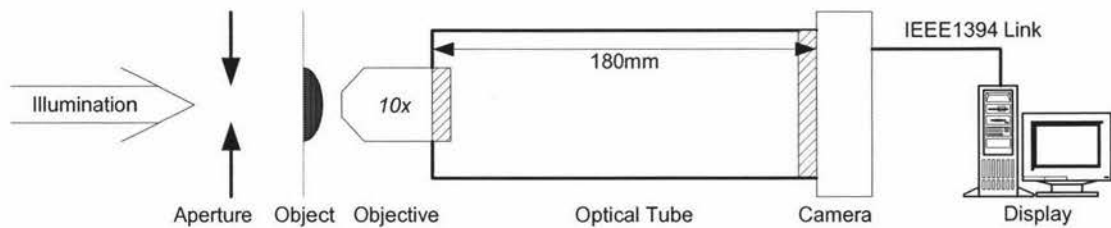
It is worth briefly mentioning the mechanical components of the design. In the experimental setup an old microscope slide table was used for x-y controls, and a translation stage driven by a screw thread was used for fine focus. Both were mounted onto an optical rail. This was shown in Figure 18. Both are bulky and impractical for a system designed to be flexible. It is important to note that this section, like section 3.1, was not researched in detail, but is rather a collection of ideas that may assist the researcher who is faced with this implementation problem.

Section 3.1 discussed potential means of presenting the specimen to the microscope. The selection of a presentation mechanism constrains the requirements for the mechanical components of the microscope. Section 3.1.1, Slide Scanning, requires the conventional attachments of a slide table with x-y controls and a means of moving the entire table in the z-direction (toward or away from the objective) for focusing. This requires considerable mechanical precision, however that technology is in widespread use. Section 3.1.2, Alternatives, discussed ideas for removing some of the axes of control that are performed by the x-y and focus controls in a conventional microscope. Removing some axes of control reduces the complexity of the mechanical controls. Also, if the system had enhanced depth of field introduced by wavefront coding (section 3.2.4), then fine focus may not be needed.

Flexibility is a theme of this research, and perhaps the most suitable design would be to install a mechanical interface on the microscope, probably around the side of the optical tube, so that presentation mechanisms designed with a known interface, and can be interchanged. The challenge would be to achieve this at low cost.



## 4.5 Final Design of a Computer Microscope



**Figure 27 Design of a Computer Microscope to optically magnify 12×**

The selected design for a computer microscope, shown in Figure 27, is much simpler than a conventional compound microscope. By replacing the human eye with a digital image sensor many simplifications to the design have been able to be made. In particular the magnification is greatly reduced, and achieved with an inexpensive standard objective. Reduced optical magnification removes the need for the complex condenser optics, and allows a cheap source of illumination to be used, such as an incandescent light bulb.

This design is more flexible than a conventional compound microscope as it is not bulky or fixed in place, making it suitable for fieldwork. Its cost is still significantly cheaper than a conventional microscope as both the conventional microscope and this microscope require a camera, and the comparative cost remains in the optics and illumination system. Importantly it stands apart from existing entry level computer microscopes such as the QX3 by providing significantly higher magnification.



## 5 Segmentation

The second block in the pollen identification system of Figure 1 is Segmentation. As noted in the introduction, segmentation involves isolating the regions within the image where there are likely to be pollen and preparing them in a suitable form for classification. It is important to note that development of the segmentation algorithm was carried out using available archive images captured on a conventional compound microscope with a camera attachment prior to the commencement of this research. This was done because the classification was developed to use those pollen images. Subsequently, as reported in Chapter 6.3.1, the segmentation algorithm was successfully trialled on pollen images captured using the computer microscope.

### 5.1 Segmentation Environment

#### 5.1.1 Constraints

In image processing we must take a pragmatic approach to problem solving. We cannot investigate all possible cases, rather capture representative images and base our algorithms on these samples. The purpose of the segmentation algorithm is to take captured images of sections of pollen slides and select individual pollen, ideally producing a set of images containing all the pollen on the slide. As with all blocks in a classification system, the segmentation process is constrained by the requirements of the processes up and down stream of it.

##### 5.1.1.1 Classification

The feature calculation module of the classification system has the input requirements summarized here:

- Images must be monochromatic
- Images must contain only one significant item (a pollen or debris), which must be surrounded by a 'plain' background.

The second of these requirements has serious implications. If overlapped pollen cannot be classified, and therefore counted, then the count is no longer accurate. The accuracy of statistical sampling is also lost as deliberately leaving out clusters of pollen that cannot be classified would significantly alter the final counts, particularly as, from observation, some pollen types tend to cluster. This has implications for the scanning system (section 3.1), where a full slide scan was

one of the techniques proposed for automating the acquisition of pollen images. If some of the pollen data is going to be ignored because it is in clusters then it is inappropriate to conduct a full slide scan. A sample is appropriate.

Given the second point above, that the square region containing segmented pollen grain may not contain any other parts of objects, then the aim of the segmentation algorithm is to segment only those pollen grains that are isolated on a clear background. Any pollen that is clustered will be immediately surrounded by other pollen and are not directly suitable for classification. These pollen cannot be completely ignored however as they are part of the sample of the environment which a slide represents. Thus it may be possible, for instance, to construct a false background around a pollen from a cluster. However isolating a pollen grain within a cluster will require a significantly different algorithm from isolating an individual pollen grain, and consequently the segmentation algorithm will initially target the isolation of individual pollen grain, as the results from this will directly satisfy the classification subsystem requirements.

#### *5.1.1.2 Capture*

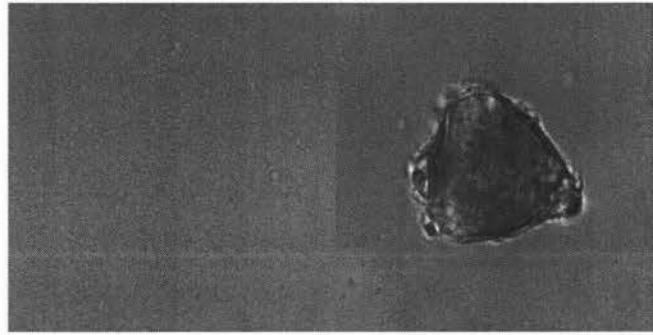
The capture subsystem provides an image of part of a pollen slide that may or may not contain pollen and detritus.

### **5.1.2 Scene Analysis**

To develop an image processing algorithm the expected contents of a scene must be known and categorized. A generic pollen slide scene is made up from seven components:

#### *5.1.2.1 Background*

The background predominates an image of a pollen slide. The background is generally very white, and does not exhibit any great or rapid changes in intensity. An example is shown on the left of Figure 28.



**Figure 28 Example images of Background and a Pollen Grain**

#### *5.1.2.2 Isolated Pollen*

Isolated pollen grains generally have a circular or round-cornered triangle shape. The appearance of pollen depends on its taxon. The region within the boundary of the pollen is generally filled with a stochastic texture (see section 5.2.2). The edges of a pollen grain are usually defined, but can be similar in intensity to the background or very thin, at times making them difficult to see. An example of a single pollen grain is shown on the right of Figure 28.

#### *5.1.2.3 Clustered Pollen*

Clustered pollen grains occur where pollen grains are overlapping and thus have the characteristics of incomplete isolated pollen grains. Two examples of clusters are shown in Figure 29.



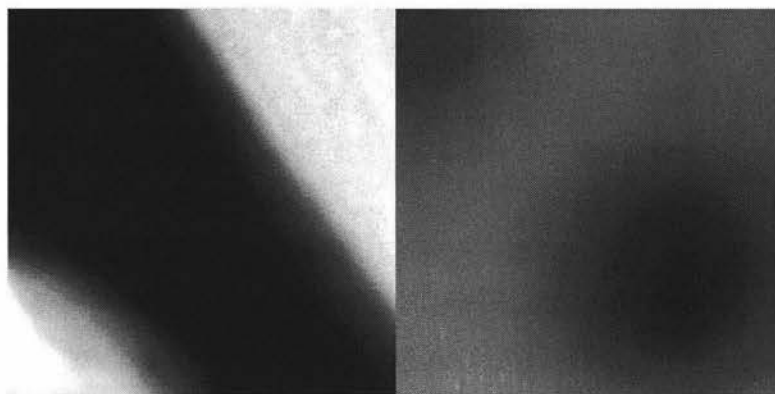
**Figure 29 Example images of Pollen Grain Clusters**

#### *5.1.2.4 Touching Pollen*

There is a transitional stage between clustered and isolated pollen, where the pollen grains are touching at a few edges but not overlapping. The left image of Figure 29 contains mostly touching pollen grains, with the exception of the top left pair, whereas the right image of Figure 29 contains completely overlapping pollen grains which could never be optically separated.

#### *5.1.2.5 Bubbles*

Bubbles are the result of air being trapped in the slide gel. These are almost always circular, with very dark and light bands. Bubbles can be larger or smaller than pollen grains. The left image in Figure 30 shows the edge of a bubble.



**Figure 30 Example images of Optical Effects**

#### *5.1.2.6 The effects of Diffraction*

These are the result of objects that are not in focus at this plane, and take the form of either Airy (mis-focus) disks or general blurs. The edges of diffractive effects are not usually well defined, and they tend to meld with the background. The right image in Figure 30 shows a diffractive spot due to a significantly out of focus object.

#### *5.1.2.7 Debris*

All other image elements are considered to be debris. Debris comes in all sizes, shapes and geometries, however some examples are more common than others. For instance 'branch-like' structures were observed to be common in the images used for development; as are small, dark, high contrast debris. Figure 31 shows several examples of debris.

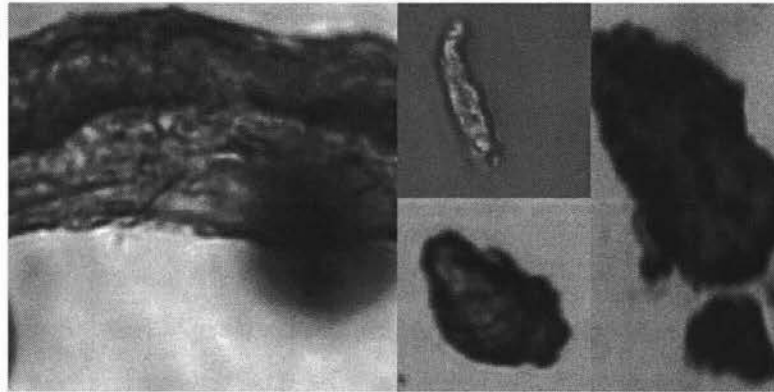


Figure 31 Example images of Debris

## 5.2 Initial Approaches

For these investigations a test set containing nine images of pollen grains was generated from a collection of compound microscope images. Each image features different pollen type and encompasses all seven elements of the scene identified in section 5.1.2 above. The pollen used differs considerably in texture and definition. Figure 32 shows three types of pollen from the test image set.



Figure 32 Three Pollen Types used in the test set (left to right): Fabaceae Sophora (2 pollen grains), Magnoliaceae Michelia doltsopa (2 pollen grains), Lamiaceae Salvia

### 5.2.1 Basic Methods

A characteristic of the pollen is that, on average, they have a different intensity to the background of the image they are part of. Therefore subtracting the image containing pollen

from an image representing the background, and applying an appropriate threshold, will reveal the positions of objects in the image. The problem is that no background images are available for this test set, and an artificial background has to be created by taking a global average. This algorithm was applied with different global thresholds. The resulting regions defined by this algorithm either contained too much background or lost significant portions of the objects. The most effective of these is included in the appendix as prog5.vip (section 7.1.1).

Pollen grains have two walls and these show clearly on optical microscopy images (section 2.1). This suggests that edge detection may be a useful approach for identifying the location of pollen grains. A variety of simple edge filters were applied to the development images. The outlines of the pollen grains were visible in the resulting images, however the edges did not form a complete loop. Without a complete loop or a reasonable approximation of one it is difficult to locate the pollen algorithmically. Additionally the edge detection methods generated significant noise.

Much of the image manipulation in the above experiments used morphological operators (section 2.5.1.2), particularly for merging disparate sections of processed pollen grains. These are the small blobs created by regional or edge processing that are closely located but are not connected. The immediately obvious approach to consolidating these into a pollen region is to use morphological closing. However there is an inherent problem in using morphological closing. In order to successfully merge the small blobs that may make up a pollen grain region a large window size is needed. The problem with a large window size is that it increases the probability of merging regions that represent different objects and therefore should not be merged. Such merging would introduce the problem of multiple objects per region, and prevents the use of simple statistical tests for screening suitable for pollen, such as a compactness test.

The basic methods applied in these experiments highlighted the complexity of the problem, in that no simple solution could be found. The most promising approach is to reduce the complexity of the problem by creating a mask that contains the pollen as well as non-pollen. This is best implemented by either the region or edge methods above. However the best results

of these experiments were not successful on all of the nine development images, most often failing on images 4 and 5.

### 5.2.2 Texture

As the basic methods have been found to be inadequate, more complex approaches are necessary. Texture has been identified as a potential technique for screening regions of the image for pollen grains because texture methods have proven successful in the classification of pollen [7]. Texture can be quantified using techniques that analyse the probability of a transition between different grey levels. There are many basic measures of texture including contrast, entropy, uniformity/energy and inertia [42-43]. Texture, as a region-based method, may be a more reliable approach as it is less sensitive to noise than an edge-based method.

A texture window filter would analyse the content of the window, performing calculations to establish the texture. If the texture content met the texture constraints of 'a pollen grain' then the region of the window would be considered part of a pollen grain. The same filter could be applied in a rejection mode. For instance bubbles have low entropy and could be eliminated in this way.

In order for this approach to be feasible, a basic textural model of a 'general' pollen grain must be identified. A texture data set was generated using a  $7 \times 7$  window passed over samples of internal pollen texture taken from the nine test images. The Energy, Homogeneity and Inertia properties were measured, using the SGLDM and equations employed in [42] with  $d=1$  and  $\text{angle}=0^\circ$ . The statistics gathered are presented in the following table:

Statistic	Minimum	Maximum	Mean
Energy	0.000432	0.126	0.00755

---

[42] R. M. Hodgson, E. J. Wood, "Texture Analysis – A New Measurement Tool For Inspection and Quality Control," in *AIM89 – Australian Instrumentation and Measurement Conference*, Adelaide, November 1989. pp270-274.

[43] Gonzalez R. C., Woods R. E., "11.3.3 Texture" in *Digital Image Processing*, 2<sup>nd</sup> ed., Prentice-Hall, New Jersey: 2001, pp.665-671



<b>Homogeneity</b>	0.0655	0.452	0.201
<b>Inertia</b>	5.07	913	160

Using a qualitative inspection, it appears that all three measures have a range of about two orders of magnitude, and are reasonably compact and balanced in ‘power-space’. Therefore these ranges can be used to determine if a window is likely to contain part of a pollen grain.

A  $7 \times 7$  textural window was passed over the image, and the textural statistics within that window calculated and compared with the general statistics. However this method has not proven successful as the complexity of the algorithm ( $O(N^6)$ ) has made it impractical to use. The source code to this is attached as an appendix, 7.1.2.

To reduce the computational complexity it may be possible to apply the texture filter after some regionalization has been done such as that suggested in section 5.2.1. The aim of applying the texture filter at this point would be to further reduce the regions within the image where pollen grains are suspected to be. However any application of this texture algorithm at this stage would still require significant processing time.

### 5.2.3 Optical Density Sub-sectioning

The Integrated Optical Density, IOD, is a measure of the average grey level within an image or a window, and is equal to the sum of pixel grey level values divided by the number of pixels [44].

Pollen will have a different IOD to the background. By setting a threshold to determine this difference locally, the edges of optically dense areas can be pinpointed. The approach taken here was to use a recursive search, starting with an area larger than a typical pollen size to identify general areas of different optical density (i.e. the presence of something other than background), then reducing the window size, searching only within areas where a larger object was believed to be.

---

[44] Imaging Research Inc., “Densitometry,” Dec 3 2003, [www.imagingresearch.com/applications/densitometry.asp](http://www.imagingresearch.com/applications/densitometry.asp)



This technique is extremely fast, however is dependent on the threshold used to select or reject blocks. This algorithm, a complicated recursion algorithm, was implemented in C++ and the software listing is attached as an appendix, 7.1.3.

#### **5.2.4 Initial Approaches: Summary**

The simplest morphological and edge-based approaches work on some images, but cannot adequately isolate only objects.

The texture approach suffers two drawbacks:

1. The values that define the statistical range of the pollen are difficult to set robustly.
2. It is impractical from the point of processor time. A small image (~40KB) takes about 12 minutes to process. A full image (~1MB) would take hours.

The IOD approach localizes and isolated the objects well for some images, and it is very fast. However it is extremely sensitive to the threshold value it uses, and unless an adaptive means were determine for setting this reliably then it is not a reliable method.

Generally none of these methods have been even close to successful across the full test image set and consequently there is need for deeper analysis.

### **5.3 Deeper Analysis**

#### **5.3.1 Test Image Statistics**

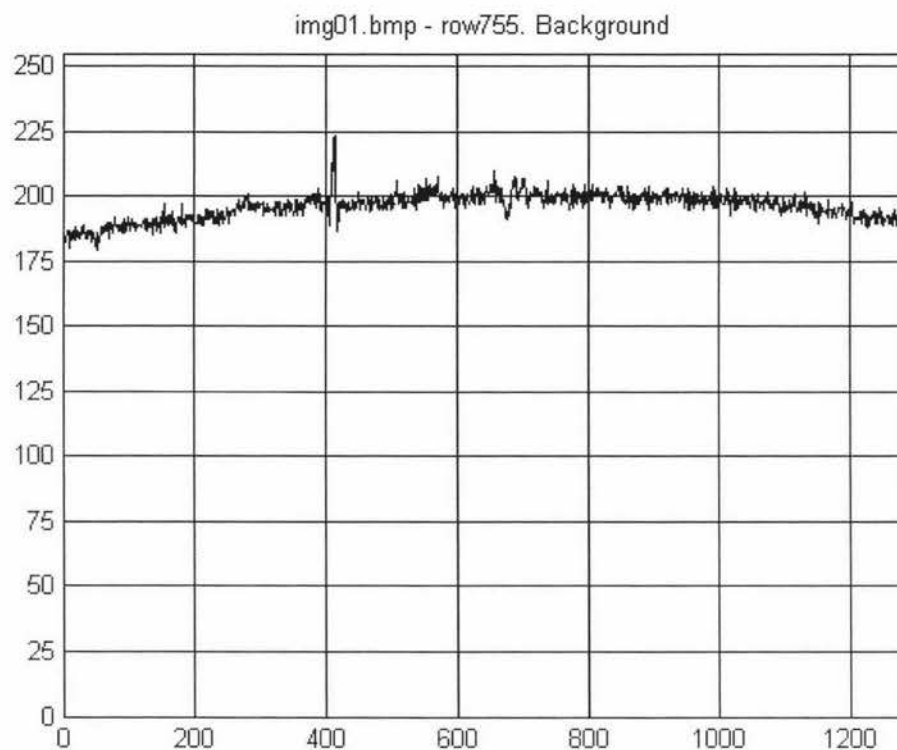
At this point it became important to conduct a more analytical survey of the test images. A total of 31 cross-sections were taken through the 9 images, each cross section containing at least one object, typically more than one. The objects of interest included pollen, out-of-focus pollen and diffraction effects, bubbles, spots, debris and background. The cross-sections were analysed for gradient by measuring the difference between adjacent pixels.

#### **5.3.2 Mean and Variance Filter**

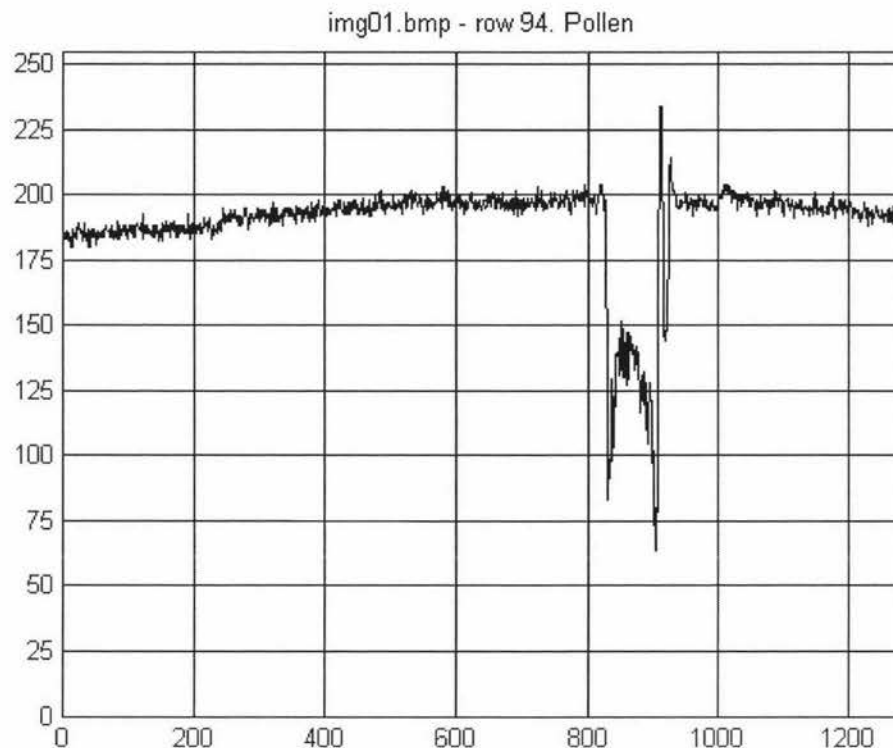
The first striking feature of the cross-sections was the characteristics of the noise. The variation in the noise is reasonably consistent across the cross-sections as shown in Figure 33 for a typical

plot, and the presence of any objects resulted in a significantly greater deviation from the mean local intensity than the noise level. Given this consistency then basic statistics are an appropriate tool for characterizing the background.

The slight parabola present over the whole image is the result of non-uniform illumination at the image capture stage. In the development images the non-uniformity was not significant enough to warrant correction, but correction may need to be considered if there was greater non-uniformity.



**Figure 33 Cross-section of background**



**Figure 34 Cross-section through a pollen grain**

The most important observation is that any objects present have intensities well outside the range of the background as shown in Figure 34. This provides a tool for constructing a 'filter'. The statistics chosen are the mean and variance. It was determined that a suitable filter would retain the pixels whose greyscale value was more than one variance level away from the mean. In practice, given the consistency of the noise variation across all test images, the background displayed above (test image 1) was used to calculate a test-set wide value for the variance. The mean was calculated as a global average for each image.

Filter summary:

$K_{ij} = 1$  where  $I_{ij} > (\mu + \text{var})$  or  $I_{ij} < (\mu - \text{var})$  where  $\mu$  is the greyscale mean of the test image, and  $\text{var} = 26.72$ , as calculated from test image 1. The Matlab function for performing this is listed in appendix 7.1.4.

This approach successfully allowed the isolation of every non-background item in the test image set. However, as the pollen grains have some regions that are similar in intensity to the

background (due to their translucency) they are not complete blobs, whereas diffraction effects, which tend to be of a low intensity gradient, are complete blobs. This can be seen in Figure 35 where the out-of-focus pollen in the top right of the original image has the most defined blob in the processed image.

The mean and variance filter was successful in identifying where objects are in these development images. However all the development images were captured under the same illumination conditions, and altering the illumination would alter the statistics of the captured images and cause the algorithm to fail to identify objects. An adaptive variance threshold would assist robustness.

#### *5.3.2.1 Alternatives*

An alternative, but similar, technique was suggested by Associate Professor Phil Bones of Canterbury University, called H-Domes (and their ‘partner’ H-Basins) [45]. This technique would probably provide similar performance to the above technique, as both require a threshold level to be set.

#### *5.3.2.2 Background Subtraction*

As noted at the beginning of the chapter, the test images that are being used for this analysis were captured prior to this research being conducted and unfortunately no independent background image is available so background subtraction was not an option. In order to integrate existing images into the classification the above mean and variance filter is required. However when integrating capture and segmentation it was found that a mean and variance filter can be replaced with a simple background subtraction and threshold, as will be detailed in chapter 6.

### **5.3.3 Edge Statistics and Enhancements**

The cross-sections also show that pollen grains have a much greater slope than non-pollen objects. The slope is the difference in greyscale level between neighbouring pixels. Treating the

---

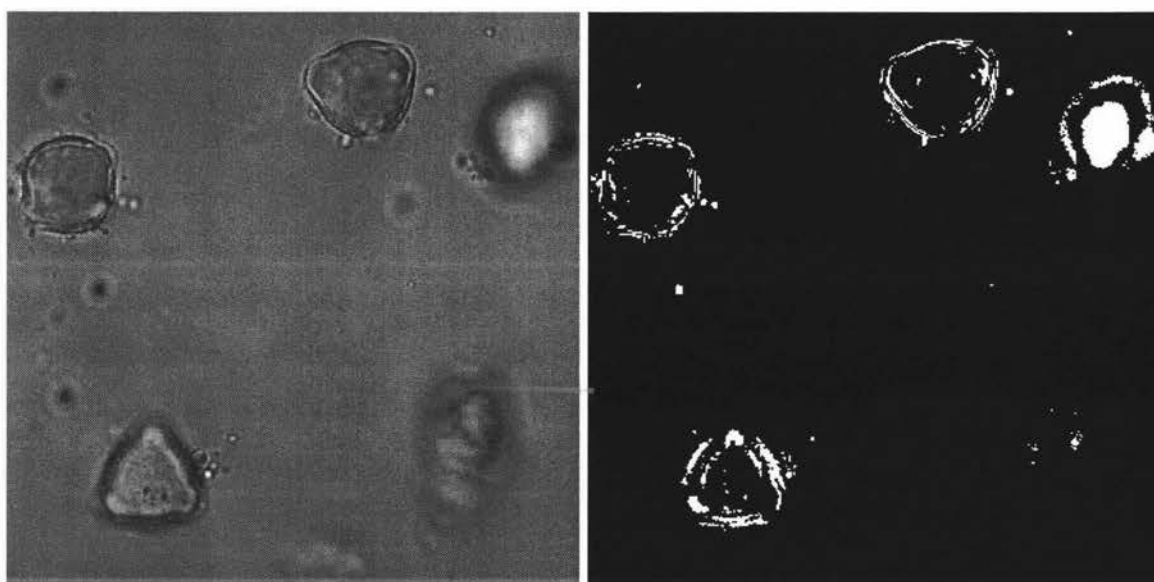
[45] L. Vincent, “Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms”, *IEEE Transactions on Image Processing*, Vol.2, No.2, April 1993

cross-sections as a sample of the whole image set, it was determined that a slope between neighbouring pixels of greater than 20 could be used as a means of identifying the edges of pollen grains. This provides a means of separating pollen grains from non-pollen objects. From the 31 cross sections it was found that if the slope between neighbouring pollen grains is greater than or equal to 20 greyscale levels then 2 false positives are generated and 1 pollen grain is missed. This is acceptable given the sampling nature of the experiment, for instance the sampled cross-sections may have missed high slopes by only one or two pixels due to local anomalies. What is important is that none of the non-pollen objects had sizable areas of high slope large enough that they would be included as a pollen grain by the slope threshold operation.

However the only region within the pollen grain that has sharp edges are the walls of the pollen. Consequently these have to be consolidated using a closing operator (9×9 window). A closing operator is not ideal for use in this algorithm, as the degree of closing required to consolidate the pollen grain region would cause neighbouring pollen grains to merge. The edges are a strong indication that a pollen is present in that region. Therefore if there exists a region in the 'mean and variance mask' image in which an edge is present, then it is likely that that region contains a pollen grain. The MATLAB function for performing the edge detection is listed in appendix 7.1.5.

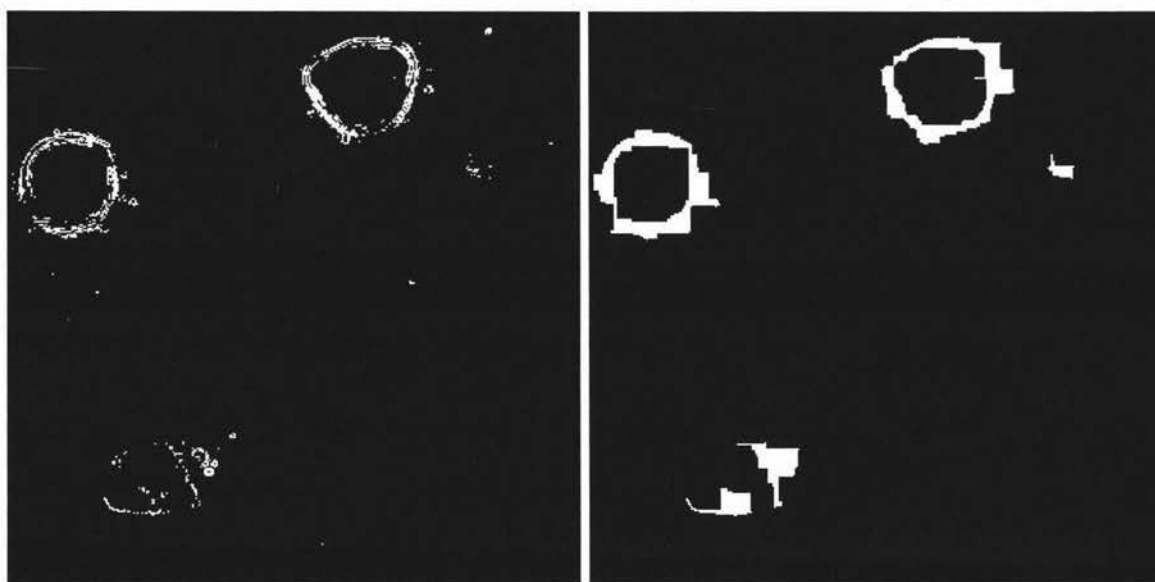
#### **5.3.4 Region Matching**

The region matching algorithm is explained below. To illustrate this an example from the development images is used. The image used was particularly difficult because of the weak edges on the pollen grains and the mixture of in focus and out of focus pollen and debris. Figure 35 shows part of the original image, and the results of applying the mean and variance mask to that image. Note that all the objects are at least partially selected by the mean and variance filter, however the strongest regions are actually the regions that have to be rejected. This example shows the need for the edge detection to select pollen grains from non-pollen objects.



**Figure 35 (left to right) Original image; image with mean and variance filter applied.**

Figure 36 shows the results of the edge detection and the consolidation of the edges. Note that in the right-hand image a small section of the out-of-focus object at the top right has survived. This will eventually be removed by thresholds that remove regions too small to be pollen grains.



**Figure 36 (left to right) image with edge detection applied; consolidated edge detection image**

Region matching is the algorithm which associates the small pockets of detected edges which indicate where pollen grains are, with the mask generated by the mean and variance filter. The

drawback in associating these images is that the detected edges represent a small part of the pollen grain, an area which is not at its centre and consequently simple operations, such as dilation, cannot be used to isolate the pollen grain area accurately.

The approach finally selected for the region matching was a simple one. Since there are two binary masks, and one should be a subset of the other, then the two masks can be AND-ed together, as demonstrated in Figure 37 (left). A pollen grain is expected to have edges on all sides. Typically at least two areas of edges on opposite sides of a pollen grain are identified by the edge detection stage. The two areas of edges are sufficient to isolate a pollen grain that is between those edges by using low-level consolidation techniques. The techniques used are: a dilation to 'complete' the outer edge ring of the pollen, a fill to fill in the pollen grain, and an erosion of equal magnitude to the previous dilation so that the original boundaries of the pollen grain are approximately restored. To complete the extraction of the pollen grain the small regions are removed. The final result is shown in Figure 37 (right).

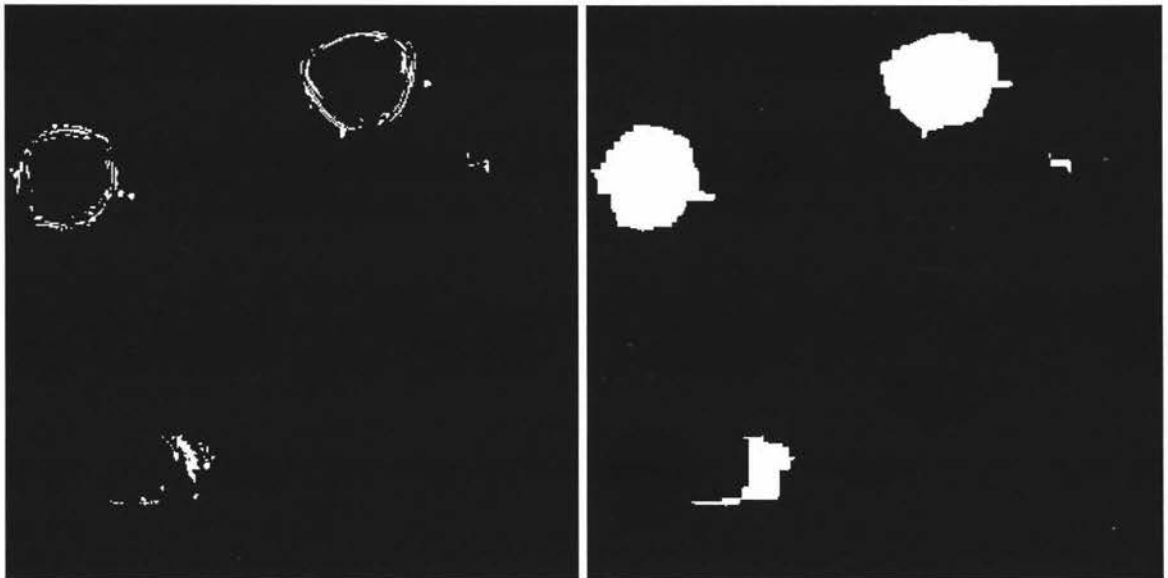
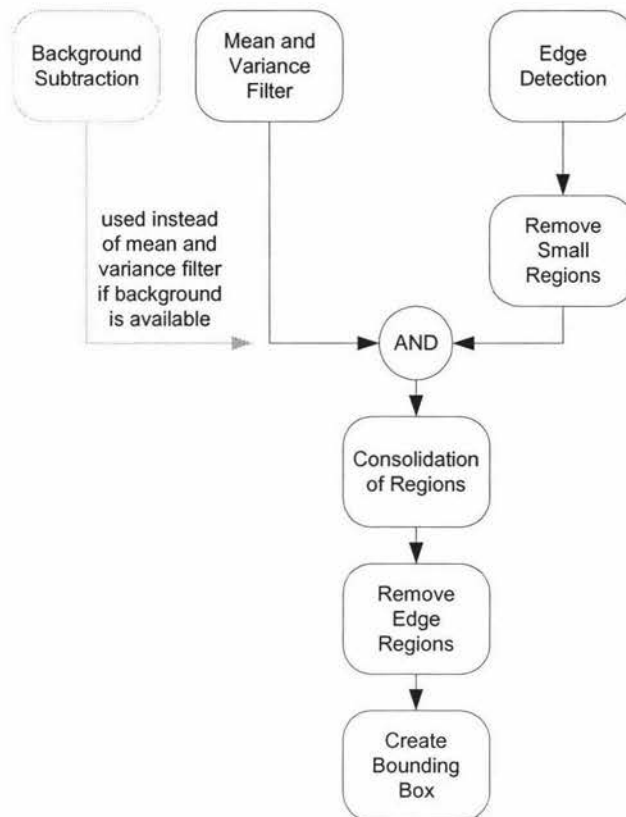


Figure 37 (left to right) image after the AND operation; image after consolidation and removal of small regions.

## 5.4 Final Algorithm

The final algorithm, mostly illustrated immediately above, has the flow chart shown in Figure 38. The full MATLAB code is listed in appendix 7.1.6.

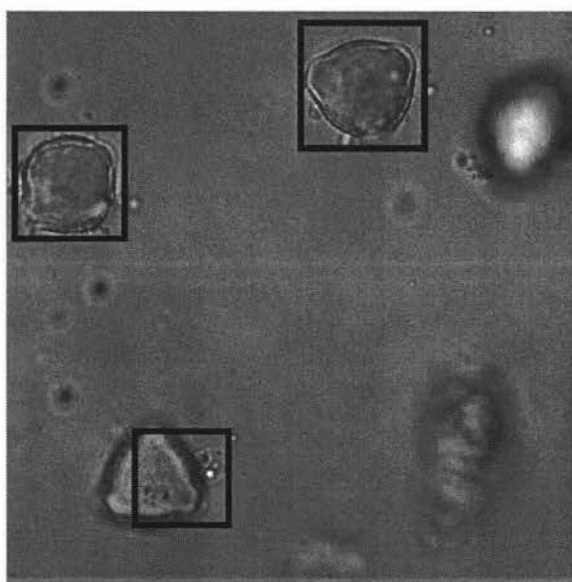


**Figure 38 Segmentation Algorithm**

This algorithm performed satisfactorily on eight of the nine test images. The major failure of this algorithm on the images within the test set is where the pollen grains are too close together and the algorithm merges these into a large blob. This is acceptable as the target is isolated pollen grains. Additionally objects that are too small to be pollen grains are being isolated. A region size threshold can easily remove these.

Figure 39 illustrates one of the problems with the segmentation: the pollen at the bottom-left of the image is only partially in focus, probably because it is sitting across several depths-of-field, and the left edges of the image of the pollen have insufficient slope for that edge to be detected. The ideal result from the segmentation would be to reject the pollen grains outright as it is not suitable for classification, however the partial but strong right and bottom edges prevent this.





**Figure 39 Image four from the development set: The final result with two pollen grains successfully identified, two large non-pollen objects rejected, and one pollen grain incorrectly identified.**

## 6 Integration

System integration is often regarded as a straightforward task by those uninitiated in the field of software engineering. Failures at the integration stage arise from problems in interfacing units that independently function correctly [46]. It is also worth noting that a 90% success rate in each unit within a system does not result in a 90% success rate for the whole system. In my experience such success rates are multiplicative, and the 10% erroneous output from each unit can cause exponential damage to the rest of the system. Thus careful integration is an important step in any software development.

This project does not involve complex software engineering, so many aspects of formal software engineering processes have been omitted. However some aspects are important in the determination of the design of the system. The user interface defines how people interact with the software and therefore constrains the design of the software. User interface considerations are considered in section 6.1. Sections 6.2 and 6.3 describe the integration process, firstly the verification of the classification algorithm, followed by the combination of the three units: capture, segmentation and classification. Section 6.4 summarizes the results of this research and recommends avenues to pursue to reach the aim of the Pollen Research Group, to automate the counting of pollen.

### 6.1 Analysis of Operator Interaction

It is expected that a pollen classification and counting system will require an operator. The operator would provide human input where it would be extremely difficult to provide automation or where the system would be greatly simplified by having operator input. The degree to which the operator interacts with the system affects the software design, as different levels of automation, and therefore user input, provide different information for the system.

#### 6.1.1 Full Automation

The ideal automated system requires no human interaction. It is ideal from a software engineer's perspective because humans introduce unknowns, are a source of errors, and are the most

---

[46] R. S. Pressman, *Software Engineering, A Practitioner's Approach*, 5<sup>th</sup> Ed., McGraw-Hill, 2001. p.488

variable factor in a system. Avoiding the uncertainties introduced by humans is a strong case for full automation. Full automation, however, is extremely difficult to achieve. Humans possess intuition and flexibility in judgement which is very difficult to program into a computer.

Consider the effects of fully automating the segmentation process. The segmentation process relies on knowing the intensity gradient (section 5.3.3) of a pollen grain for determining if a region is likely to contain pollen. The gradient threshold for selecting pollen is determined by sampling the difference between neighbouring pixels and comparing the results for pollen grains and non-pollen objects. Thus to find the gradient threshold automatically requires knowing where the pollen grains are, which in turn requires knowing the gradient threshold, and gives rise to a 'Hole in the Bucket' scenario [47]. Furthermore, under full automation, the segmentation process would have to work with a wide range of illumination, low contrast, and non-uniform illumination. Designing and testing an algorithm to cater for all potential cases is verging on the impossible.

Additionally, from an economic perspective, full automation, at this time, is not necessary. In current microscopy an operator is needed to load and focus the specimen slides, so requiring a human to assist in the operation of some parts of the processing will not necessitate expensive additional staff.

### **6.1.2 Calibration by Example**

There is an operator interaction continuum ranging from full automation and no operator interaction at one end to no automation and the operator manually analysing the slides at the other. Full automation has been shown to be currently impractical, so an automation solution must be found which is near to full automation on the continuum, but with concessions made to operator interaction where it is more practical to do so.

The example in the previous section illustrated the problem of needing to know where pollen are to determine some global conditions of the system. An operator could provide this information

---

[47] Referring to the Scouts' song *There's a Hole in My Bucket*, which illustrates a cycle of requirements where in order to repair the hole in the bucket it is eventually shown that we require the bucket.

by, say, drawing a box around a pollen grain in the captured image, from which information can be gathered. This provides data unique to that situation, both in terms of the capture conditions such as the illumination, and in terms of the characteristics of a particular taxon of pollen present. This approach is not without its faults. Firstly, the assumption has been made that the first pollen grain found is representative of the remainder. This will be the case frequently, as pollen grains have been shown to exhibit differentiating characteristics from other objects on the slide, but not universally. Secondly, this approach relies on the correctness of the operator, as there would be no straightforward means of checking. If there were, an operator probably wouldn't be required!

Operator interaction will also allow a background image to be captured separately, a step which saves considerable complexity by removing the mean and variance filter as shown in Figure 38. A background image also allows analysis of the illumination conditions under which the images are captured. The information gathered from this analysis can be used to correct for illumination faults.

### **6.1.3 Summary**

The process where an operator identifies and isolates a pollen grain from a captured image by drawing a box around it has been selected as the level of operator interaction. The pollen isolated by this process will henceforth be referred to as the 'reference pollen'.

## **6.2 Classification**

As stated in section 1.2.2, the classification sub-system is the result of the research carried out by Zhang[8]. Zhang's research demonstrated a 92% success rate in classifying 16 different taxa of pollen. The classification itself was performed by a neural network. The data used to train the network and the data input for classification are vectors describing an image of a pollen grain by its size, shape, and textural features.

It is important to confirm that the classification subsystem performs as expected before it is integrated into the full system. The images used to test the classification are the same as those used by Zhang, and a classification rate of 91% has been achieved. This is similar to the 92%

achieved by Zhang, where the discrepancy between the results is due to the weights of the neural network being initialised with random values. Some of the source code for the classification is attached as an appendix (section 7.1.7). The training and testing systems associated with the classification are summarized diagrammatically in Figure 40. For further detail, Zhang’s report is recommended [8].

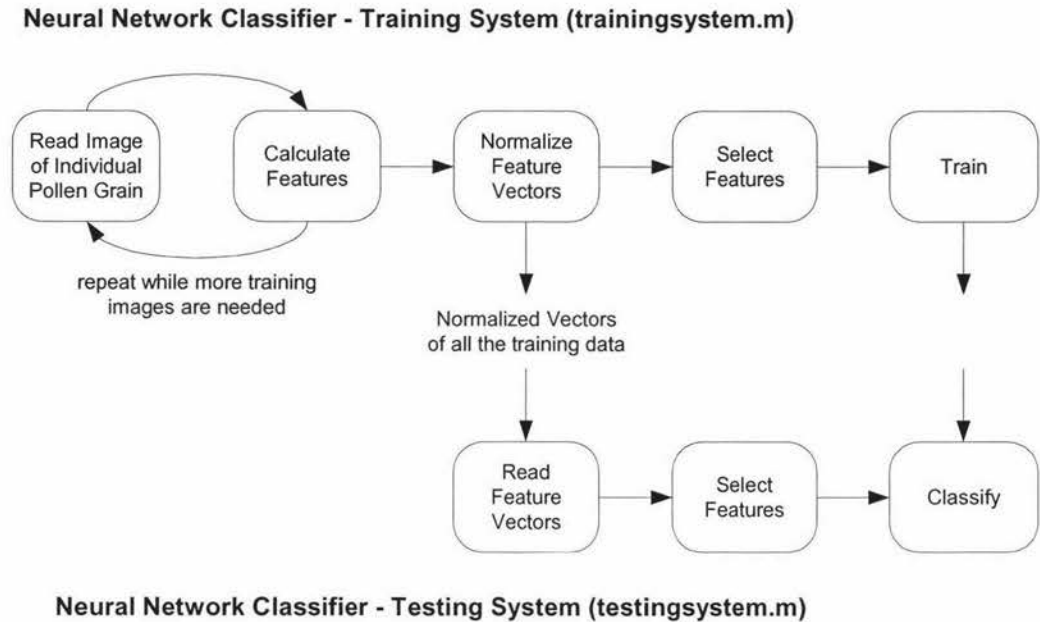
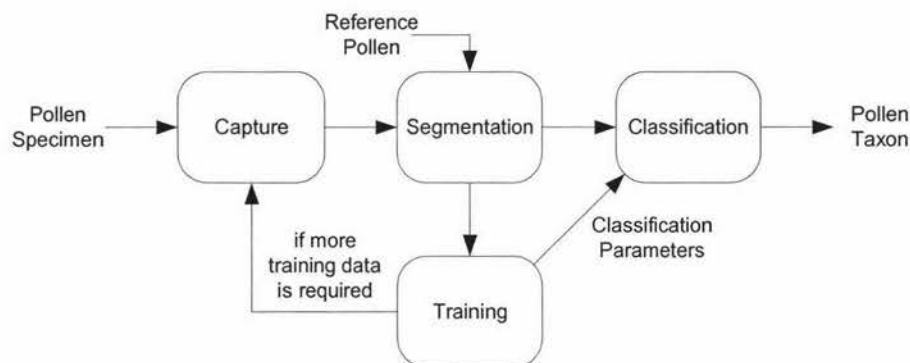


Figure 40 Classification Sub-System

### 6.3 Integration

There are two paths through the full system, which is shown in Figure 41. The first of these is the training system, which gathers a sufficient number of pollen images of each taxon so that a neural network can be trained to identify that genus in the presence of other pollen. The second of these paths is the classification of a single pollen grain using the classification parameters established by the training system to classify the pollen.



**Figure 41 Automatic Pollen Classification System**

The complete source code for the integration is listed in sections 7.1.7 and 7.1.8.

### 6.3.1 Testing

To confirm that the integrated system is functioning correctly testing is required. The appropriate test for this system is to capture images of pollen slides using the computer microscope, segment them, train a neural network using the segmented images, and then test the whole system by processing segmented images through the classification. The test set used for this testing consists of approximately 70 images captured using the computer microscope, intended to contain 120 individual pollen images, 30 of each taxon. 16 of each taxon will be used to train the neural network, and all 30 of each taxon will be used to test the neural network.

Unfortunately enough time had passed between the development of the capture process and the integration that some of the pollen that had originally been successfully captured, appear to have died or dehydrated and become opaque. As most of the pollen had become less transparent the camera was set to saturate the image so that the contrast of the pollen was enhanced. The benefit of doing this is that the images captured, of pollen and debris on a completely white background, were more comparable to the images that had been used during the development of the classification, than the images previously captured using the computer microscope.


Initial failures in the system occurred in the segmentation process. The four taxa used in the integration testing varied in size significantly from the large *Pinus radiata* to the small *Rumex obtusifolius*. The fixed thresholds in the segmentation used for rejecting regions based on their size were not suitable for these extremes and adaptive thresholds were introduced. The values

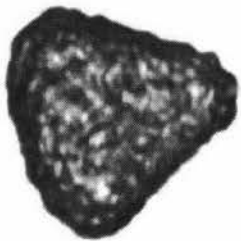
for these size thresholds were based on the area of the reference pollen. A second failure was the rejection of a large number of pollen grains in the images based on their proximity to other pollen grains or debris. This was corrected by introducing a further subdivision algorithm that separates a local cluster of objects into individual regions and ignores close neighbours.


## 6.3.2 Results


### 6.3.2.1 Segmentation

To confirm that the segmentation process was functioning correctly, the images containing pollen grains were segmented. The threshold values here are included for future reference as to what might be expected for these values. Visual inspection has confirmed the number of pollen grains present should be at least approximately 30 for each taxa. The errors record regions which should not have been present in the segmented image set.

<b>Betula pendula</b>		 <p><b>Figure 42 Betula pendula</b></p>
Gradient Threshold	43	
RegionArea Threshold	4278	
RegionSelectLower Threshold	1283	
Image Count	31	
Errors	1	

<b>Cyathea dealbata</b>		 <p><b>Figure 43 Cyathea dealbata</b></p>
Gradient Threshold	29	
RegionArea Threshold	24436	
RegionSelectLower Threshold	7331	
Image Count	31	
Errors	0	

<b>Pinus radiata</b>		 <p><b>Figure 44 Pinus radiata</b></p>
Gradient Threshold	30	
RegionArea Threshold	31080	
RegionSelectLower Threshold	9324	
Image Count	24	
Errors	0	

<b>Rumex obtusifolius</b>		 <p><b>Figure 45 Rumex obtusifolius</b></p>
Gradient Threshold	33	
RegionArea Threshold	6156	
RegionSelectLower Threshold	1846	
Image Count	51	
Errors	5	

The image count for Pinus radiata is a little low. This is due to the elongated shape of some pollen grains, which has caused a failure of in the compactness test. Increasing the compactness threshold subtly corrects this problem without introducing too many errors elsewhere.

The five errors for Rumex obtusifolius are due to two pollen grains being almost directly on top of each other. The segmentation system was designed to separate objects with the general size and shape characteristics of pollen grains, and two pollen grains almost completely on top of each other have these characteristics and therefore cannot be rejected at segmentation. Thus the five errors are not failures in terms of the design specifications of the segmentation, but may not be classified correctly.



#### 6.3.2.2 Pollen Classification System

Images containing 64 individual pollen grains, 16 of each taxon, were used to train the neural network. A total of 120 images containing pollen were then used to test the neural network. The results are as follows:

Pollen Correctly Classified	115
Pollen Incorrectly Classified	5
Success Rate	96%

Two failures occurred where *Betula pendula* were incorrectly classified as *Rumex obtusifolius*, and three failures occurred where *Rumex obtusifolius* were incorrectly classified as *Betula pendula*.

## 6.4 Conclusions and Recommendations

At each stage pleasing results have been obtained, however all of the modules in the system could be improved and there are areas where completely new solutions could enhance the performance of the prototype. The system as it stands is essentially a working prototype and a suitable basis for system improvement.

### 6.4.1 Capture

The computer microscope developed during this thesis can produce images at a resolution and of a quality comparable to advanced and expensive compound optical microscopes. By my estimates, the computer microscope is also about a tenth of the price of the microscopes its performance was compared with. However to be effective the computer microscope needs some improvements. Firstly the operating controls and environment of the prototype microscope was not satisfactory. The microscope was difficult to focus and difficult to capture still images from because it was mechanically unstable and therefore exhibited wobble. Wobble is alleviated in some microscopes by the use of a granite bench, however once again this is expensive, and more economic solutions may be possible while retaining the flexibility of the computer microscope.

At this stage, the major gap in the automation of pollen counting is the collection and presentation of the pollen to the microscope. This was considered in section 3.1. It should be noted that using servomotors to automate the control of a slide, while removing a laborious task for the operator, does not meet the aim of automation as the slides still have to be prepared. The alternatives suggested in section 3.1.2 identify means of eliminating the slide preparation stage and are worth some consideration.

Finally the effect of the illumination on the system is not well understood. It is known that microscope objectives with a high numerical aperture, which increases with magnification, require correct Köhler illumination. The computer microscope does not have high optical magnification, so simple illumination such as an incandescent light has been used. Although this illumination appears adequate, it may not be the optimum.

#### **6.4.2 Segmentation**

The segmentation is the most complex aspect of the system. This is the result of the difficulty in segmenting images where the objects have similar intensities to the background, and a result of the evolving nature of the development of the segmentation module. The results of the segmentation are excellent, achieving the segmentation of isolated pollen and pollen that are in close proximity but are not touching each other. The segmentation algorithm was never intended to be perfect, ideally the classification module will identify unknown objects, and set them aside for manual classification or rejection. Thus the segmentation module meets the standards it was designed to meet.

Speaking with a colleague recently I suggested that the ideal image processing system should use no fixed thresholds. Whether this can be justified or not I do not know, however I am certain that making some of the thresholds in the segmentation adaptive, based on the reference pollen, was the key factor in making the segmentation system operational. However there are still thresholds in the early processing which are not adaptive, such as those which reject non-pollen objects on the basis of size, and while these did not appear to create failures, they remain an inflexibility in the system. Additionally the limit of the ability of the system to adapt to pollen

grains of different shapes and sizes is not completely known, and future pollen grains outside of these limits may or may not cause problems.

Finally neither the segmentation nor classification can deal with tightly clustered pollen. Touching pollen however are possible to work with, provided an artificial white background is constructed around them as dictated by the requirements for the classification module. Such a separation of touching pollen using region-based processing could be included in the subdivide function (see section 7.1.8.6).

### **6.4.3 Classification**

Zhang's[8] research was able to classify 16 pollen taxa, and this full system segmented and classified four taxa. Both of these numbers are sufficient to prove the concept that a system can be implemented to automatically classify pollen, but neither take into consideration the important real-world matter of classifying unknown objects. At some stage an 'unknown' class needs to be introduced into the neural network training data so that objects which are not pollen grains, or not pollen grains known to the system can be rejected, or set aside for manual investigation. Manual analysis of unknown objects has the additional benefit of enabling the system, which includes the operator, to find the 'one in a thousand' pollen that Stillman and Flenley identified as desirable [4].

There are improvements to be made to the classification system. During this research twelve vectors have been used in the classification. Zhang selected these twelve vectors on a pragmatic 'trial and error' basis. An analysis method needs to be introduced so that the best features can be selected for classifying the pollen. This experimental design problem is shortly to be investigated by the Pollen Research Group. Furthermore the neural network training itself needs to be more flexible, and the source code for training and classifying the neural network may benefit from being replaced by library functions which would allow the parameters to be set more flexibly.

### **6.4.4 User Interface**

The user interface does not appear in any system diagrams for a reason: so far there is no user interface. However to be a practical and marketable solution, an automated pollen counting

system will require a user-friendly interface. MATLAB does not lend itself to the design of sophisticated user interfaces so another language, such as C# or Delphi which have a better mix of user interface design tools and low level mathematical operators, should be used.

#### **6.4.5 Conclusion**

This project necessitated the capture of detailed images of pollen grains having diameters of between 15 $\mu$ m and 100 $\mu$ m, with feature sizes in the order of 1 $\mu$ m or more. Early work on the project demonstrated experimentally and theoretically that whereas a single lens microscope is not possible, a three lens system is like to be sufficient when using a digital image sensor. In practice the use of a standard microscope objective and no other lenses was demonstrated to be cost-effective.

Tests on the final version of the integrated system described resulted in a rate of 96% success for the classification of the selected pollen. In the tests the pollen were presented on prepared slides containing both pollen and detritus. The system performed image capture, using the computer microscope, and segmentation prior to classification performed using Zhang's software. It is concluded that these results clearly demonstrated that an automated pollen counting and classification system is an attainable goal. The final stage to be developed is a system for sampling and presentation of airborne and/or ancient pollen.

## 6.5 References

A numerical listing of reference is given here. The ‘missing’ numbers are purely footnotes and are not relevant in a reference section.

- [1] C. A. Holdaway, R. M. Hodgson, “Reinventing the Microscope in the Age of Digital Imaging” in *Image and Vision Computing New Zealand*, 2003, pp.286-290.
- [3] P. Li, J .R. Flenley, L. K. Empson, “Classification of 13 types of New Zealand pollen patterns using neural networks,” in *Proc. International Conference on Image and Vision Computing*, Auckland, 1998, pp. 120-123.
- [4] E. C. Stillman, J. R. Flenley, “The Needs and Prospects for Automation in Palynology,” *Quaternary Science Reviews*, vol. 15, pp. 1–5, 1996.
- [5] W. J. Treloar, *Digital Image Processing Techniques and their Application to the Automation of Palynology*, PhD Thesis, University of Hull, Hull, Great Britain.
- [6] W. J. Treloar, *Automation of Palynology using Image Processing and Pattern Recognition Techniques*, Postdoctoral Research Report, Massey University, 1994.
- [7] M. Forster, J. Flenley, “Pollen Purification and Fractionation by Equilibrium Density Gradient Centrifugation,” *Palynology*, 17(1993), pp137-155.
- [8] Y. Zhang, *Pollen Discrimination Using Image Analysis*, Postdoctoral Research Report, Massey University, New Zealand, 2003.
- [9] J. Emberlin , B. Adams-Groom , “What is Pollen,” <http://pollenuk.worc.ac.uk/aero/pm/WIP.htm>
- [10] T. E. Weier, M. G. Barbour, C. R. Stocking, T. L. Rost, *Botany*, Singapore: John Wiley & Sons, 1982, pp286-295.
- [11] R. B. Knox, *Pollen and Allergy*, Southhampton: Edward Arnold Publishers,1979, pp3-8, 22-24.
- [12] D. W. Fountain, “Pollen and Inhalant Allergy,” *Biologist*, vol. 49, no. 1, 2002, pp.5-9.
- [13] H. D. Young, R. A. Freedman, *University Physics*, Reading, Mass: Addison-Wesley, 1996, pp.1053-1192.
- [14] B. Walker, *Optical Engineering Fundamentals*, McGraw-Hill Inc., 1995, pp.50-53.
- [15] F. Pedrotti, L. Pedrotti, *Introduction to Optics*, Englewood Cliffs, N.J.: Prentice-Hall International, 1993.

- [16] F. Pedrotti, L. Pedrotti, *Introduction to Optics*, Englewood Cliffs, N.J.: Prentice-Hall International, 1993, pp.126-129.
- [17] B. H. Walker, "7.3 A Typical Lens Application," in *Optical Engineering Fundamentals*, McGraw-Hill, 1995, pp.150-154.
- [18] "Microscope", Encyclopaedia Britannica. Encyclopaedia Britannica Premium Service. 2 Mar 2003. [www.britannica.com](http://www.britannica.com)
- [19] E. M. Slayter, H. S. Slayter, "The Light Microscope," in *Light and Electron Microscopy*, Cambridge University Press, 1992, p.131-148.
- [21] S. W. Paddock, T. J. Fellers, M. W. Davidson, "Introduction to Confocal Microscopy" [www.microscopyu.com/articles/confocal/confocalintrobasics.html](http://www.microscopyu.com/articles/confocal/confocalintrobasics.html)
- [22] How Stuff Works, "What is the difference between CCD and CMOS image sensors in a digital camera?," Jan 2004. <http://www.howstuffworks.com/question362.htm>
- [23] Gonzalez R. C., Woods R. E., "10 Segmentation" in *Digital Image Processing*, 2<sup>nd</sup> ed., New Jersey: Prentice-Hall, 2001, pp.567-642.
- [24] J. R. Flenley, "The problem of Pollen Recognition", in M. B. Clowes and J. P. Penny, *Problems in Picture Interpretation*, pp141-145. CSIRO, Canberra, 1968.
- [25] J. R. Flenley, P. Li, L. K. Empson, "Identification of 13 Pollen Types by Neural Network Analysis of Texture Data Only," in *Proc. Of Image and Vision Computing New Zealand*, 1999, pp.295-298. 621-367 IVC
- [26] I. France, A. W. G. Duller, G. A. T. Duller, H. F. Lamb, "A new approach to automated pollen analysis," *Quaternary Science Reviews*, vol.19, 2000, pp.537-546.
- [27] E. M. Slayter, H. S. Slayter, *Light and Electron Microscopy*, Cambridge University Press, 1992.
- [28] L. Jelinek, G. Peters, J. Okuley, S. McGowan "Dissection of the Intel® Play™ QX3™ Computer Microscope," *Intel Technology Journal*, Q4, 2001.
- [29] Olympus, "Olympus MIC-D Product Information", <http://www.mic-d.com/product/spec.html>, 12 Jan. 2004.
- [30] S. C. Tucker, W. T. Cathey, E. R. Dowski, "Extended depth of field and aberration control for inexpensive digital microscope systems," *Optics Express*, vol. 4, no. 11, pp. 467-474, 24 May 1999.

- [31] Olympus Corp., "BX51 Research System Microscope Brochure", [www.olympus.com](http://www.olympus.com), January 2004.
- [32] T. N. Cornsweet, *Visual Perception*, New York: Academic Press Inc., 1970. pp.7-9.
- [33] M. Born, E. Wolf, "5.3 The primary (Seidel) aberrations" in *Principles of Optics*, 4<sup>th</sup> Ed., Pergamon Press. 1970. pp.211-218.
- [37] Warren J. Smith, *Modern Optical Engineering*, McGraw-Hill, 1966, Chapters 3 and 10, pp.49-71 and 247-279.
- [38] F. L. Pedrotti. L. S. Pedrotti, "16-2 Beam Spreading" in *Introduction to Optics*, 2nd Ed. New Jersey: Prentice-Hall, 1993, pp.329-330.
- [39] W. J. Smith, "The Design of Optical Systems" in *Modern Optical Engineering*, USA. McGraw-Hill, 1966. pp.340-347.
- [40] Edmund Optics, Optics and Optical Instruments Catalog, New Jersey. 2002. Online copy available at [www.edmundoptics.com](http://www.edmundoptics.com)
- [42] R. M. Hodgson, E. J. Wood, "Texture Analysis – A New Measurement Tool For Inspection and Quality Control," in *AIM89 – Australian Instrumentation and Measurement Conference, Adelaide*, November 1989. pp.270-274.
- [43] Gonzalez R. C., Woods R. E., "11.3.3 Texture" in *Digital Image Processing*, 2nd ed., Prentice-Hall, New Jersey: 2001, pp.665-671.
- [44] Imaging Research Inc., "Densitometry," Dec 3 2003, [www.imagingresearch.com/applications/densitometry.asp](http://www.imagingresearch.com/applications/densitometry.asp)
- [45] L. Vincent, "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms", *IEEE Transactions on Image Processing*, Vol.2, No.2, April 1993.
- [46] R. S. Pressman, *Software Engineering, A Practitioner's Approach*, 5<sup>th</sup> Ed., McGraw-Hill, 2001. p.488.



## 7 Appendices

### 7.1 Source Code

#### 7.1.1 Basic Methods

This program is the last of a series that investigated the potential of creating artificial backgrounds and using simple morphological operators. This is written in VIPS, kindly provided by Dr Donald Bailey.

```
program 01

! This relies too heavily on consistent background illumination.
! With more generously set thresholds it could be a regionalizer.

    delete p5_*
    clear
    set auto /on

    declare string p5_filename

    let p5_filename = %string(#1) & ".bmp"
    load /bmp 'p5_filename' p5_orig

    invert p5_orig

    box average p5_orig p5_smooth_orig 3x3
    box average p5_orig p5_background 127x127

    stat p5_smooth_orig 0 255 p5_dummy p5_mean1 p5_sd1
    stat p5_background 0 255 p5_dummy p5_mean2 p5_sd2

    let p5_th1 = %integer(p5_mean1) + %integer(p5_sd1) + %integer(p5_sd1)
    let p5_th2 = %integer(p5_mean2) + %integer(p5_sd2) + %integer(p5_sd2)

    threshold p5_smooth_orig 'p5_th1'
    threshold p5_background 'p5_th2'

    or p5_smooth_orig p5_background
    and p5_smooth_orig p5_orig

end
```

#### 7.1.2 Texture Filtering

Similar code was used to determine the texture statistics. This code applies the texture filter to the image.



The bmp.h and bmp.cpp files are not included as they are implementations which I coded of a standard OS/2 or Windows bitmap.

#### 7.1.2.1 main.cpp

```
#include <stdio.h>
#include <conio.h>
#include <math.h>

#include "texturefilter.h"
#include "bmp.h"

int main(void)
{
    char szFilename[] = "a01a.bmp";
    const int Ng = 256, angle = 0;

    CTextureFilter txtfilt(Ng);
    CBmp * pBmp;
    byte ** img = 0;
    int height = 0, width = 0;

    pBmp = new CBmp();
    if(pBmp->readFile(szFilename) != 0)
    {
        printf("File %s could not be read\n", szFilename);
        getch();
        delete pBmp;
        return -1;
    }

    //Current Pollen Image Data and Palette
    byte ** oldImg = pBmp->getIMGpal();
    RGBQUAD * oldPalette = pBmp->getPalette();

    //generate new image and set it black (so edge pixels are set);
    BITMAPINFOHEADER bmi = pBmp->getInfoHeader();
    width = bmi.biWidth;
    height = bmi.biHeight;

    //note: newIMG contains intensities, not palette entries.
    byte ** newImg = new byte*[height];
    for(int i=0; i<height; i++)
    {
        newImg[i] = new byte[width];
        for(int j=0; j<width; j++) newImg[i][j] = 0;
    }

    //windowing - 7x7
    byte ** window = new byte*[7];
    for(int i=0; i<7; i++) window[i] = new byte[7];

    double energy, homoge, inert;

    //set pixel to 1 if it contains pollen like texture within a window.
```

```

for(int i=3; i<(height-3); i++)
{
    for(int j=3; j<(width-3); j++)
    {
        for(int di=-3; di<=3; di++)
            for(int dj=-3; dj<=3; dj++)
                window[3+di][3+dj] =
                    oldPalette[ oldImg[i+di][j+dj] ].rgbBlue;

        txtfilt.generateSGLDM(window, 7, 7, angle);
        txtfilt.normalizeSGLDM(7, 7, angle);

        energy = txtfilt.getEnergy();
        homoge = txtfilt.getHomogeneity();
        inertl = txtfilt.getInertia();

        if((energy > 0.00432) && (energy < 0.125) &&
            (homoge > 0.06551) && (homoge < 0.452) &&
            (inerti > 5.07457) && (inerti < 913.0))
        {
            newImg[i][j] = 255;
        }
        else
        {
            newImg[i][j] = 0;
        }

        printf(".");
    }
    printf("%d",i);
}

//newImg now contains 255 where texture is good, 0 elsewhere.
//creating a new image, largely copying old one.
BITMAPFILEHEADER bfh = pBmp->getFileHeader();
BITMAPINFOHEADER bih = pBmp->getInfoHeader();
int paletteSize = 1 << bih.biBitCount;
RGBQUAD * bmpPalette = new RGBQUAD[paletteSize];
for(int i=0; i<paletteSize; i++)
{
    bmpPalette->rgbBlue = i;
    bmpPalette->rgbGreen = i;
    bmpPalette->rgbRed = i;
}

//reusing pointer :)
delete pBmp;
pBmp = new CBmp();
pBmp->setFileHeader(bfh);
pBmp->setInfoHeader(bih);
pBmp->setPalette(bmpPalette, paletteSize);
pBmp->setIMGpal(newImg, height, width);
pBmp->writeFile("a0lamask.bmp");

for(int j=0; j<height; j++) delete [] img[j];
delete [] img;
img = 0;

```

```

    printf("%s processed\n", szFilename);

    printf("Complete.  Press any key.\n");
    getch();
    return 0;
}

```

### 7.1.2.2 *texturefilter.h*

```

typedef unsigned char byte;

//note: only distance of 1 (d=1) is programmed :)
//angle: is a choice of 0, 45, 90, 135.  Anything else gets rejected.

//CTextureFilter does not copy the image or the palette.
//the only memory it owns is the SGLDM.
class CTextureFilter
{
public:
    CTextureFilter(int Ng);
    ~CTextureFilter();

    //sgldm methods
    void generateSGLDM(byte ** img, int Nx, int Ny, int ang);
    void normalizeSGLDM(int Nx, int Ny, int ang);

    //output methods
    double getEnergy();
    double getHomogeneity();
    double getInertia();
    void printMatrix(FILE * f_out);

private:
    int generate0(byte ** img, int Nx, int Ny, int i, int j);
    int generate45(byte ** img, int Nx, int Ny, int i, int j);
    int generate90(byte ** img, int Nx, int Ny, int i, int j);
    int generate135(byte ** img, int Nx, int Ny, int i, int j);

    double normalize0(int Nx, int Ny);
    double normalizeDiag(int Nx, int Ny);
    double normalize90(int Nx, int Ny);

    double ** _SGLDM; //SGLDM matrix
    int _Ng;           //Number of graylevels
};

```

### 7.1.2.3 *texturefilter.cpp*

```

#include <stdio.h>
#include <math.h>
#include "texturefilter.h"

CTextureFilter::CTextureFilter(int Ng)
{
    //set graylevels
    _Ng = Ng;
}

```

```

        //allocate _SGLDM:
        _SGLDM = new double*[_Ng];
        for(int a=0; a<_Ng; a++) _SGLDM[a] = new double[_Ng];
    }

CTextureFilter::~CTextureFilter()
{
    //destroy _SGLDM
    for(int a=0; a<_Ng; a++) delete [] _SGLDM[a];
    delete [] _SGLDM;
}

//----- SGLDM METHODS -----
void CTextureFilter::generateSGLDM(byte ** img, int Nx, int Ny, int ang)
{
    //generating S(i,j,1,0)
    int count = 0;
    for(int i=0; i<_Ng; i++)
    {
        for(int j=0; j<_Ng; j++)
        {
            switch(ang)
            {
                case 0:
                    count = generate0(img, Nx, Ny, i, j);
                    break;
                case 45:
                    count = generate45(img, Nx, Ny, i, j);
                    break;
                case 90:
                    count = generate90(img, Nx, Ny, i, j);
                    break;
                case 135:
                    count = generate135(img, Nx, Ny, i, j);
                    break;
                default:
                    printf("Invalid Angle. Use 0, 45, 90 or 135\n");
            }

            _SGLDM[i][j] = (double)count;
        }
        //printf("%d ",i);
    }
}

void CTextureFilter::normalizeSGLDM(int Nx, int Ny, int ang)
{
    double normFact = 1;
    switch(ang)
    {
        case 0:
            normFact = normalize0(Nx, Ny);
            break;
        case 45:

```

```

        case 135:
            normFact = normalizeDiag(Nx, Ny);
            break;
        case 90:
            normFact = normalize90(Nx, Ny);
            break;
        default:
            printf("Invalid Angle. Use 0, 45, 90 or 135\n");
    }

    for(int i=0; i<_Ng; i++)
        for(int j=0; j<_Ng; j++)
            _SGLDM[i][j] /= normFact;
}

//----- OUTPUT METHODS -----
double CTextureFilter::getEnergy()
{
    double energy = 0.0;
    for(int i=0; i<_Ng; i++)
        for(int j=0; j<_Ng; j++)
            energy += (_SGLDM[i][j] * _SGLDM[i][j]);
    return energy;
}

double CTextureFilter::getHomogeneity()
{
    double homogeneity = 0.0;
    for(int i=0; i<_Ng; i++)
        for(int j=0; j<_Ng; j++)
            homogeneity += (_SGLDM[i][j] / (1+(i-j)*(i-j)));
    return homogeneity;
}

double CTextureFilter::getInertia()
{
    double inertia = 0.0;
    for(int i=0; i<_Ng; i++)
        for(int j=0; j<_Ng; j++)
            inertia += (_SGLDM[i][j] * (i-j)*(i-j));
    return inertia;
}

void CTextureFilter::printMatrix(FILE * f_out)
{
    for(int x=0; x<_Ng; x++)
    {
        for(int y=0; y<_Ng; y++)
            fprintf(f_out, "%f ", _SGLDM[x][y]);
        fprintf(f_out, "\n");
    }
    fprintf(f_out, "\n");
}

//----- PRIVATE METHODS -----

```

```

int CTextureFilter::generate0(byte ** img, int Nx, int Ny, int i, int j)
{
    //heading: right, left
    int count = 0;
    for(int x=0; x<Nx; x++)
        for(int y=0; y<Ny-1; y++)
        {
            if((img[x][y] == i) && (img[x][y+1] == j)) count++;
            if((img[x][y] == j) && (img[x][y+1] == i)) count++;
        }
    return count;
}

int CTextureFilter::generate45(byte ** img, int Nx, int Ny, int i, int j)
{
    //heading: down-left, up-right
    int count = 0;
    for(int x=0; x<Nx-1; x++)
        for(int y=1; y<Ny; y++)
        {
            if((img[x][y] == i) && (img[x+1][y-1] == j)) count++;
            if((img[x][y] == j) && (img[x+1][y-1] == i)) count++;
        }
    return count;
}

int CTextureFilter::generate90(byte ** img, int Nx, int Ny, int i, int j)
{
    //heading: down, up
    int count = 0;
    for(int x=0; x<Nx-1; x++)
        for(int y=0; y<Ny; y++)
        {
            if((img[x][y] == i) && (img[x+1][y] == j)) count++;
            if((img[x][y] == j) && (img[x+1][y] == i)) count++;
        }
    return count;
}

int CTextureFilter::generate135(byte ** img, int Nx, int Ny, int i, int j)
{
    //heading: down-right, up-left
    int count = 0;
    for(int x=0; x<Nx-1; x++)
        for(int y=0; y<Ny-1; y++)
        {
            if((img[x][y] == i) && (img[x+1][y+1] == j)) count++;
            if((img[x][y] == j) && (img[x+1][y+1] == i)) count++;
        }
    return count;
}

double CTextureFilter::normalize0(int Nx, int Ny)
{
    //normalization factor  $R(1,0) = 2Ny(Nx-1)$ 
    return 2*(double)Ny*((double)Nx-1);
}

```

```

double CTextureFilter::normalizeDiag(int Nx, int Ny)
{
    //normalization factor  $R(1,45) = R(1,135) = 2(Nx-1)(Ny-1)$ 
    return 2*((double)Ny-1)*((double)Nx-1);
}

double CTextureFilter::normalize90(int Nx, int Ny)
{
    //normalization factor  $R(1,90) = 2Nx(Ny-1)$ 
    return 2*(double)Nx*((double)Ny-1);
}

```

### 7.1.3 Optical Density

As for the texture filter, bmp.h and bmp.cpp files are not included as they are implementations which I coded of a standard OS/2 or Windows bitmap.

```

#include <stdio.h>
#include <conio.h>
#include <string.h>
#include "bmp.h"

const double IODthold = 180.0;

byte ** newImage(int height, int width)
{
    byte ** img = new byte*[height];
    for(int i=0; i<height; i++)
    {
        img[i] = new byte[width];
        //initialization to 0
        for(int j=0; j<width; j++) img[i][j] = 0;
    }
    return img;
}

void deleteImage(byte ** img, int height)
{
    for(int i=0; i<height; i++)
        delete [] img[i];
    delete [] img;
}

//input: img
//output: imgOut
void pass1(byte ** img, int dim, int vertOffset, int horzOffset,
int imgHeight, int imgWidth, byte ** imgOut)
{
    if(dim < 8)
    {
        for(int i=vertOffset; i<imgHeight; i++)
            for(int j=horzOffset; j<imgWidth; j++)
                imgOut[i][j] = 255;
        return;
    }
}

```

```

    }

    int sum; double IOD;
    for(int i=vertOffset; i<=(imgHeight-dim); i+=dim)
    {
        for(int j=horzOffset; j<=(imgWidth-dim); j+=dim)
        {
            //window average
            sum = 0;
            for(int di=0; di<dim; di++)
                for(int dj=0; dj<dim; dj++)
                    sum += img[i+di][j+dj];
            IOD = (double)sum / (double)(dim*dim);

            if(IOD < IODthold)
                pass1(img, dim/2, i, j, i+dim, j+dim, imgOut);
        }
    }
}

void pass2(byte ** img, int dim, int vertOffset, int horzOffset, int
imgHeight, int imgWidth, byte ** imgOut)
{
    //pass 2 needs to start such that the bottom-right
    //square of size dim*dim, sits in the bottom right
    //of the area searched
    int nVertBoxes = imgHeight / dim; //integer division.
    int nHorzBoxes = imgWidth / dim;

    int height = nVertBoxes * dim;
    int width = nHorzBoxes * dim;

    vertOffset = imgHeight - height;
    horzOffset = imgWidth - width;

    pass1(img, dim, vertOffset, horzOffset, height, width, imgOut);
}

//OR. Bitwise. Range of 0-255
void imgOR(byte ** img1, byte ** img2, byte ** imgOut, int height, int width)
{
    for(int i=0; i<height; i++)
        for(int j=0; j<width; j++)
            imgOut[i][j] = img1[i][j] | img2[i][j];
}

//AND. Bitwise. Range of 0-255
void imgAND(byte ** img1, byte ** img2, byte ** imgOut, int height, int
width)
{
    for(int i=0; i<height; i++)
        for(int j=0; j<width; j++)
            imgOut[i][j] = img1[i][j] & img2[i][j];
}

int main(void)
{

```



```

char szFilename[256];
char szTemp[256];

//assuming paletted (256) for now :)
CBmp bmp;

printf("Enter file name to analyse:");
gets(szFilename);

if(bmp.readFile(szFilename) != 0)
{
    printf("File not found\n");
    getch();
    return -1;
}

BITMAPFILEHEADER bmpFileHeader = bmp.getFileHeader();
BITMAPINFOHEADER bmpInfoHeader = bmp.getInfoHeader();
int height = bmpInfoHeader.biHeight;
int width = bmpInfoHeader.biWidth;
RGBQUAD * bmpPalette = bmp.getPalette();
byte ** oldImg = bmp.getIMGpal();

//allocate temp images
byte ** img = newImage(height, width);
byte ** pass1Mask = newImage(height, width);
byte ** pass2Mask = newImage(height, width);
byte ** finalMask = newImage(height, width);

//copy intensities from oldImg -> img
for(int i=0; i<height; i++)
    for(int j=0; j<width; j++)
        img[i][j] = bmpPalette[ oldImg[i][j] ].rgbBlue;

//replace palette with standard grayscale:
int paletteSize = 1 << bmpInfoHeader.biBitCount;
bmpPalette = new RGBQUAD[paletteSize];
for(int i=0; i<paletteSize; i++)
{
    bmpPalette[i].rgbBlue = i; bmpPalette[i].rgbGreen = i;
    bmpPalette[i].rgbRed = i; bmpPalette[i].rgbReserved = 0;
}
bmp.setPalette(bmpPalette, paletteSize);

//region dimension
const int dim = 64;

//Two-pass Integrated Optical Density Search
pass1(img, dim, 0, 0, height, width, pass1Mask);
pass2(img, dim, 0, 0, height, width, pass2Mask);

//Image Processing Ops
imgOR(pass1Mask, pass2Mask, finalMask, height, width);
imgAND(finalMask, img, img, height, width);

//Copy back and Save
bmp.setIMGpal(img, bmpInfoHeader.biHeight, bmpInfoHeader.biWidth);

```

```

sprintf(szTemp, "out_%s", szFilename);
bmp.writeFile(szTemp);

//delete pointers
deleteImage(img, height);
deleteImage(pass1Mask, height);
deleteImage(pass2Mask, height);
deleteImage(finalMask, height);

printf("Done. Press a key to end.\n");
getch();
return 0;
}

```

### 7.1.4 Mean and Variance Filter

The remainder of the code is written in MATLAB.

```

function img = variancemask(img, mean, var)
%function img = variancemask(img, mean, var)
%applies the mean and variance filter to input img.

[xdim,ydim] = size(img);

for x=1:xdim
    for y=1:ydim
        if (img(x,y) > (mean + var)) | (img(x,y) < (mean - var))
            img(x,y) = 255;
        else
            img(x,y) = 0;
        end
    end
end
end

```

### 7.1.5 Edge Detection

```

function [iout,ivd,ihd] = edgescreen(img)
%function iout = edgescreen(img)
%Returns edges in img whose gradient is greater or equal to 20.

imgHt = size(img,1);
imgWt = size(img,2);

img = double(img);

%create vertical difference (gradient) image
ivd = img(2:imgHt,:) - img(1:(imgHt-1),:);

%create horizontal difference (gradient) image
ihd = img(:,2:imgWt) - img(:,1:(imgWt-1));

%if grad >= 20 then keep :)
ivd = abs(ivd) >= 20;
ihd = abs(ihd) >= 20;

```

```
%make square and logically OR
ivd = [ivd ; zeros(1,imgWt)];
ihd = [ ihd' ; zeros(1,imgHt) ]';

iout = ivd | ihd;
```

### 7.1.6 Complete Segmentation Algorithm

```
%runscript: general stuff
clear all
region_thres = 50;
region_thres_two = 200;

for i=1:9

    img = imread(['img0', int2str(i), '.bmp']);
    [ydim, xdim] = size(img);

    %Apply mean and variance filter to the image to create main mask
    i1 = variancemark(img, mean(mean(img)), 26.72);
    %apply edge method to original image
    i2 = edgescreen(img);

    %consolidate regions
    i3 = imclose(i2,ones(13,13));

    %remove small regions
    [L,n] = bwlabel(i3,8);

    for a=1:n
        [r,c] = find(L==a);
        if(size(r,1) < region_thres)
            i3(r(1:size(r,1)),c(1:size(r,1))) = 0;
        end
    end

    %combine masks.
    i4 = and(i1,i3);
    %consolidate - close with a fill
    i5 = imdilate(i4,ones(29,29));
    i6 = imfill(i5,'holes');
    i7 = imerode(i6, ones(29,29));

    %remove regions touching the edges of the image
    i8 = removeEdgeRegions(i7);

    %remove small regions
    [L,n] = bwlabel(i8,8);
    for a=1:n
        [r,c] = find(L==a);
        if(size(r,1) < region_thres_two)
            i8(r(1:size(r,1)),c(1:size(r,1))) = 0;
        end
    end
end
```

```

%too lazy to find out the sizes :)
finalmask = i8*0;

[L,n] = bwlabel(i8,8);
%find statistics
centstats = regionprops(L,'Centroid');
axisstats = regionprops(L,'MajorAxisLength');
for a=1:n
    %create a box about the centroid using the radius
    cent = [centstats(a).Centroid];
    r = [axisstats(a).MajorAxisLength];
    %q => [centroid row, centroid height]
    cen_c = cent(1,1);
    cen_r = cent(1,2);

    %bounding box: (for now)
    ul_c = round(cen_c - (r/2));
    ul_r = round(cen_r - (r/2));

    %place onto finalmask
    for i=0:r-1
        for j=0:r-1
            xcoord = ul_c + i;
            ycoord = ul_r + j;
            %check point is within image.
            if((xcoord > 0) && (xcoord <= xdim) && (ycoord > 0) && (ycoord <= ydim))
                finalmask(ycoord, xcoord) = 1;
            end
        end
    end

end

%overlay and output
%i9 = immultiply(logical(finalmask), img); %straight overlay.. hides
%rest of image

%draws a box around objects masked.
f1 = imdilate(finalmask, ones(9,9));
f2 = imsubtract(f1, finalmask);
f3 = imcomplement(f2);

i9 = immultiply(f3, im2double(img));

figure, imshow(i9);

end

```

#### 7.1.6.1 *removeEdgeRegions Function*

```

function iout = removeEdgeRegions(img)
%function iout = removeEdgeRegions(img)
%img must be binary
%removes regions which touch the edges of the image, and would therefore be
%only part of an object.

```

```

%create a positive mask that contains only 2 rows of edge.
[ydim,xdim] = size(img);

mask = zeros(ydim,xdim);
vertedge = ones(ydim,2);
horzedge = ones(2,xdim);

mask(:,1:2) = vertedge;
mask(:,xdim-1:xdim) = vertedge;
mask(1:2,:) = horzedge;
mask(ydim-1:ydim,:) = horzedge;

%combine mask with binary img.
ix1 = and(mask, img);

%At this stage the only objects will be regions who have some pixels within
%a two pixel width border.
%get a list of coords of all white points
[r,c] = find(ix1);

if(size(r)<1)

%use select to find the regions in the img (check that we haven't already
%0'd it)
    i2 = bwselect(img,c,r,8);
    i3 = xor(ones(ydim,xdim),i2); %invert i2
    iout = and(img, i3);          %and with original image, thus removing the
regions                                     %we had selected from the original img.
else
    iout = img;
end

```

## 7.1.7 Training and Testing Systems (v2.01)

### 7.1.7.1 TrainingSystem.m

```

% This m-file is the high-level script for training a neural network to
% classify pollen images.
% To update the classifier the input images should be changed here. Running
% the script will then generate new a1 a2 diffman and collectmean .mat
% files which are the variables required to normalize and lclassify a pollen
% image.
%
% Craig Holdaway. Last updated 12 2 04 Version 1.99

clear all

%select images to be procssed for training - this would be changed for
%different images
subdirectories = char('BP', 'CD', 'PR', 'RO');
rootpath = 'H:\integration - micro seg';

%adjust the size of this to suit the number of vectors
number_of_images = 4*30;

```

```

features = zeros(number_of_images,46);

for i=1:4 %4 = number of subdirectories
    path = [rootpath '\' deblank(subdirectories(i,:)) ];
    fprintf(1,'Processing %s', deblank(subdirectories(i,:)) );

    for j=1:30 %30 = number of image files to process in each subdirectory
        if j <= 9
            suffix = ['0' int2str(j)];
        else
            suffix = int2str(j);
        end

        img = imread([path '\a' suffix '.bmp'] , 'bmp');
        features(30*(i-1) + j, :) = polFeatureCalculation(img); %30 = total
j.

        fprintf(1, '.');
    end
    fprintf(1, '\n');
end

%normalize
out = polNormalize(features);

save out out

%training with a reduced data set of 16 of each pollen type and 12
%features to give a training set of size 256x12
trainset = [];
trainset = [trainset; out( 1: 16, :) ; out( 31: 46, :) ; out( 61: 76, :) ;
out( 91:106, :)];
trainset = [trainset; out( 1: 16, :) ; out( 31: 46, :) ; out( 61: 76, :) ;
out( 91:106, :)];
trainset = [trainset; out( 1: 16, :) ; out( 31: 46, :) ; out( 61: 76, :) ;
out( 91:106, :)];
trainset = [trainset; out( 1: 16, :) ; out( 31: 46, :) ; out( 61: 76, :) ;
out( 91:106, :)];

trainset2 = [trainset(:,1), trainset(:,2), trainset(:,4), trainset(:,5),
trainset(:,18), trainset(:,20), ...
            trainset(:,24), trainset(:,25), trainset(:,32), trainset(:,35),
trainset(:,41), trainset(:,46)];

polTraining(trainset2');

7.1.7.2 TestingSystem.m

% System for testing the correctness of the trained neural network using
% all of the reference pollen normalized during TrainingSystem.m
%
% Craig Holdaway. Last updated 12 2 04 Version 1.99

clear all
%normalized pollen vectors: 480 x 46
load out out

```

```

testset = [out(:, 1), out(:, 2), out(:, 4), out(:, 5), out(:,18), out(:,20),
            out(:,24), out(:,25), out(:,32), out(:,35), out(:,41), out(:,46)];
testset = testset';

correct = 0; incorrect = 0;
for i=1:120
    %NN
    load a1 a1 %weight matrix
    load a2 a2 %weight matrix
    test_hidden1 = sig(testset(:,i),a1);
    test_output = sig(test_hidden1,a2);

    k = find(test_output==max(test_output));

    %outputs: 1 = 5 = 9 = 13, 2 = 6 = 10 = 14. etc.
    k = mod(k,4);
    if(k==0)
        k = 4;
    end

    %stats:
    expected_result = ceil(i/30);
    if(expected_result == k)
        correct = correct + 1;
    else
        incorrect = incorrect + 1;
        fprintf(1, 'Result = %d. Expected Result = %d\n',k,
expected_result);
    end
end

fprintf(1, 'Correct = %d\n', correct);
fprintf(1, 'Incorrect = %d\n', incorrect);

```

#### 7.1.7.3 *polFeatureCalculation.m*

This function is not my work. It has been extracted directly from Zhang's source code. See the attached CD for the full listing. Below is the interface and first few lines of the source.

```

function vFeature = polFeatureCalculation(img)
%function vFeature = polFeatureCalculation(img)
%Calculates a feature vector from the image 'imagenamename'
% Parameters:
%     imagenamename: A string containing the path of the image to process
% Return Value:
%     vFeature = Feature Vector representing the input image. Size = 46x1
%
% Dr Yongping Zhang. Last updated 12 2 04 Version 1.99
if nargin ~= 1
    error('An image is required')
end

if isa(img, 'uint8') | isa(img, 'uint72')
    img = im2double(img);
end

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%Pre-processing of images: segmentation and changing the grey level

```

```

M = size(img,1); % obtain the size of image
N = size(img,2);
I=ones(M,N);
...

```

#### 7.1.7.4 *polNormalize.m*

```

function out = polNormalize(polVectors)
% function out = polNormalize(polVectors)
% Normalises feature vectors based on previous normalization data.
% Parameters:
%     P: Matrix of vectors to be normalized. Size = [width,46];
% Return Value:
%     out: Normalized vectors. Same size as input.
%
% Dr Yongping Zhang and Craig Holdaway. Last updated 12 2 04 Version 2.01

[nVectors, nFeatures] = size(polVectors);

if nFeatures ~= 46
    error('Feature vectors should be 46 columns wide.');
```

end

```

collectionMean=mean(polVectors);
collectionMax=max(polVectors);
collectionMin=min(polVectors);

acolumn = ones(nVectors,1);
meanFeatures = [];
for i=1:nFeatures
    meanFeatures = [meanFeatures, collectionMean(i).*acolumn];
end

%diff: absolute difference between the feature value for each pollen and
%the feature value for that feature.
diff = abs(polVectors - meanFeatures);
diffmax = max(diff);

%difference between the maximum value and the mean for each feature. 1x46
save diffmax46 diffmax;
%mean value of each feature. 1x46
save collectmean46 collectionMean;

maximumDifferences=[];
for i=1:46
    maximumDifferences = [maximumDifferences, diffmax(i).*acolumn];
end

% normalize the feature vectors
out=(polVectors - meanFeatures) ./ maximumDifferences;

```



#### 7.1.7.5 *polTraining.m* and *sig.m*

These functions are not my work. They have been reproduced directly from Dr Zhang's source code. See the attached CD for the full listing. Below is the interface and comments to the source.

```
function polTraining(P)
% function polTraining(P)
% training a 12-25-16 MLP
%
% Dr Yongping Zhang. Last updated 12 2 04 Version 1.99

function y = sig(A,W)
%function y = sig(A,W)
% compute the output of NN for given inputs A and weight matrix W
%
% Dr Yongping Zhang. Last updated 2003 Version 1.99
```

### 7.1.8 Complete Integration Algorithm (v2.02)

Some of the low level functions which are repeated unchanged from the Training and Testing systems are not included here. See section 7.1.7 immediately above.

#### 7.1.8.1 *Runscript.m*

This is the high level function which controls the segmentation and classification processes. It is specific to the four pollen used to test the integrated system.

```
function runscript(dir)
%function runscript(dir)
% dir is a string with the directory name.
%Use '.' for the current directory.
%
% Craig Holdaway. Last updated 24 2 04 Version 2.02
% Derived from scriptsplit.m in version 2.00

%Defaults: Some are modified later.
CDF_Percentage_Cut = 96; %slope cut needed for cst. See cst.m for details
grad_thres = 20; %a default value - calling cst will overwrite it
RegionSizeLowerThreshold = 3000; %actual region not square enclosing region

%overwriting arg for now.
dir = 'H:\integration - micro seg\RO';

%Get slope threshold
filename = [dir, '\01.bmp'];
img = imread(filename);
[grad_thres, dummy, regionarea] = cst(img, CDF_Percentage_Cut);
fprintf('Threshold = %d\n', grad_thres);
fprintf('RegionArea = %d\n', regionarea);

RegionSizeLowerThreshold = regionarea * 0.3
```

```

%Open background file
fileback = [dir, '\background.bmp'];
img_bgnd = imread(fileback);

count = 1;
%Process the Images
for j=1:13
    %Read image from file
    if(j < 10)
        jfile = ['0', int2str(j)];
    else
        jfile = int2str(j);
    end
    filename = [dir, '\', jfile, '.bmp'];

    img = imread(filename);
    %Process Image. Arg four: 2, so a masked image is returned.
    img_regionized = processImage(img, img_bgnd, grad_thres, 2);

    %if some regions containing pollen have been selected.
    if( max(max(img_regionized)) > 0)
        img_binary = im2bw(img_regionized,1/256);
        img_labeled = bwlabel(img_binary,4); %4-connectivity

        %classification data struct. class_coords
        %format: [ topleft_col topleft_row width height; ... ]
        class_coords = [];

        %Perform checks on each region
        total_regions = max(max(img_labeled));
        for i=1:total_regions
            %count the corners
            i10 = double(img_labeled == i); %select one region
            h = [ 1 1 ; 1 1 ];
            i11 = imfilter(i10, h);
            i12 = (i11 == 1);
            corner_count = sum(sum(i12));

            if(corner_count >= 4)
                coords = subdivide(i10, img,
                                   RegionSizeLowerThreshold);
                class_coords = [class_coords ; coords];
            %else corner count < 4 then on edges so discard.
            end
        end

        %Classify
        [r,c] = size(img);
        for i=1:size(class_coords,1)
            %create region from coords.
            %format: [ topleft_col topleft_row width height;...]

            topleft_col = class_coords(i,1) -20;
            topleft_row = class_coords(i,2) -20;
            width = class_coords(i,3) +40;
            height = class_coords(i,4) +40;
        end
    end
end

```

```

%check the image edges haven't been exceeded
if(topleft_col <= 0)
    topleft_col = 1;
end
if(topleft_row <= 0)
    topleft_row = 1;
end

while(topleft_col+width > c)
    width = width - 2;
end
while(topleft_row+height > r)
    height = height - 2;
end

region = img(topleft_row : topleft_row + height -1,
            topleft_col : topleft_col + width -1);

region_processed = removeEdgeRegions2(region,
                                     RegionSizeLowerThreshold);

result = OnePollenClassify(region_processed)
%imwrite(result, [dir '\a' int2str(count) '.bmp']);

count = count + 1;

end

end % (max(max(img_regionized)) > 0)
fprintf('processed\n');
end

```

#### 7.1.8.2 Cst.m

```

function [result, distn, regionarea] = cst(img, CDF_Percentage_Cut)
%function [result, distn, regionarea] = cst(img, CDF_Percentage_Cut)
%
% This function generates the slope distribution within the image img, and
% returns the cut-slope for a single-pixel difference edge detector.
%
%At this stage there still exists the problem that the GUI interaction is
%limited. For instance if there were no (good) pollen on the first slide
%then we'd have a problem... Also it would be nice to get data from
%multiple pollen. Once everything else works then I'll consider (the pain
%of) using the MATLAB GUI tools.
%
% Craig Holdaway. Last updated 13 2 04 Version 2.00

if((CDF_Percentage_Cut > 100) || (CDF_Percentage_Cut < 0))
    error('CDF_Percentage_Cut must be between 0 and 100');
end

%msgbox(['Draw a diagonal line across a pollen in the image (that is about to
appear).' ...
%       'Press enter when both points have been entered.' ...
%       'This should have ends at the opposite vertices of a tight-
fitting box.'], 'Pollen Identikit', 'help')

```

```

[BW,xi,yi] = roipoly(img);
if((size(xi,1) < 2) || (size(yi,1) < 2) )
    error('Polygon must have at least two verticies');
end

%establish the square
x1 = round(min(xi));
y1 = round(min(yi));
x2 = round(max(xi));
y2 = round(max(yi));

%third return val.
regionarea = (x2-x1)*(y2-y1);

%process rows
itr = 1;
diffsR = zeros(y2-y1, x2-x1);
for r = y1:y2
    rowR = double(img(r, x1:x2));
    width = size(rowR,2);
    diffsR(itr,:) = rowR(1, 2:width) - rowR(1, 1:width-1);
    itr = itr + 1;
end

%process cols
itr = 1;
diffsC = zeros(x2-x1, y2-y1);
for c = x1:x2
    colC = double(img(y1:y2, c));
    height = size(colC,1);
    diffsC(itr,:) = (colC(2:height, 1) - colC(1:height-1, 1))';
    itr = itr + 1;
end

%The elements in diffsC and diffsR are the slope between adjacent pixels
%We need a distribution of these statistics.
distrn = zeros(256,1);
%note offset of 1:
%i.e. row 1 is the count of how many times 0 was the slope,
%i.e. row 2 is the count of how many times 1 was the slope, etc.
[r,c] = size(diffsR);
for x=1:r
    for y=1:c
        distrn( abs(diffsR(x,y))+1, 1) = distrn( abs(diffsR(x,y))+1, 1) + 1;
    end
end

[r,c] = size(diffsC);
for x=1:r
    for y=1:c
        distrn( abs(diffsC(x,y))+1, 1) = distrn( abs(diffsC(x,y))+1, 1) + 1;
    end
end

%find the Xth percentile index:
cnt = sum(distrn);

```

```

cutpoint = cnt * CDF_Percentage_Cut/100;

itr = 1;
CDF = 0;
while(CDF < cutpoint)
    CDF = CDF + distn(itr);
    itr = itr + 1;
end

%itr is the index of the slope value above the cutpoint.
%Recalling there is an offset of 1, then we treat itr as the slope value
%which will be used for the edge detector
result = itr;

```

### 7.1.8.3 *processImage.m*

```

function i9 = processImage(img, img_bgnd, grad_thres, for_display)
%function i9 = processImage(img, img_bgnd, grad_thres)
% img is the image to process
% img_bgnd is the background image
% grad_thres is for edge detection
% for_display selects an output format: 0 for binary mask of pollen areas
%                                     1 for boxes drawn
%
around pollen areas
%
% Craig Holdaway. Last updated 13 2 04 Version 2.00

%minimum intensity difference between an object and the background
backgnd_sub_thresh = 60;
%screens for removing regions which are too small to be pollen
region_thres = 50;
region_thres_two = 200;
%minimum dimension for the final bounding box
bounding_box_thold = 50;

[ydim, xdim] = size(img);

%background subtraction
%img_bgnd - img as background is lighter than objects of interest
i0 = imsubtract(img_bgnd,img);

%threshing
i1 = im2bw(i0,backgnd_sub_thresh/256);

%apply edge method to original image
i2 = edgescreen(img, grad_thres);

%consolidate regions
i3 = imclose(i2,ones(13,13));

%remove small regions
[L,n] = bwlabel(i3,8);

for a=1:n
    [r,c] = find(L==a);
    if(size(r,1) < region_thres)
        i3(r(1:size(r,1)),c(1:size(r,1))) = 0;
    end
end

```

```

        end
    end

    %combine masks.
    i4 = and(i1,i3);
    %consolidate - close with a fill
    i5 = imdilate(i4,ones(29,29));
    i6 = imfill(i5,'holes');
    i7 = imerode(i6, ones(29,29));

    %remove regions touching the edges of the image
    i8 = removeEdgeRegions(i7);

    %remove small regions
    [L,n] = bwlabel(i8,8);
    for a=1:n
        [r,c] = find(L==a);
        if(size(r,1) < region_thres_two)
            i8(r(1:size(r,1)),c(1:size(r,1))) = 0;
        end
    end

    %too lazy to find out the sizes :)
    finalmask = i8*0;

    [L,n] = bwlabel(i8,8);
    %find statistics
    centstats = regionprops(L,'Centroid');
    axisstats = regionprops(L,'MajorAxisLength');
    for a=1:n
        %create a box about the centroid using the radius
        cent = [centstats(a).Centroid];
        r = [axisstats(a).MajorAxisLength];

        if(r > bounding_box_thold)

            cen_c = cent(1,1);
            cen_r = cent(1,2);

            %bounding box: (for now)
            ul_c = round(cen_c - (r/2));
            ul_r = round(cen_r - (r/2));

            %place onto finalmask
            for i=0:r-1
                for j=0:r-1
                    xcoord = ul_c + i;
                    ycoord = ul_r + j;
                    %check point is within image.
                    if((xcoord > 0) && (xcoord <= xdim) && (ycoord > 0)
                        && (ycoord <= ydim))
                        finalmask(ycoord, xcoord) = 1;
                    end
                end
            end
        end
    end
end

```

```

end

if(for_display == 0)
    %binary mask
    i9 = finalmask;
elseif(for_display == 1)
    %draws a box around objects masked.
    f1 = imdilate(finalmask, ones(9,9));
    f2 = imsubtract(f1, finalmask);
    f3 = imcomplement(f2);

    i9 = immultiply(f3, im2double(img));
elseif(for_display == 2)
    %original image within mask
    i9 = immultiply(logical(finalmask), img);
else
    error('for_display argument is invalid. Select 0 for binary
        mask; 1 for display boxes');
end

```

#### 7.1.8.4 *EdgeScreen.m*

Called by *runscript.m* to select the edges in the image. See section 7.1.5.

#### 7.1.8.5 *removeEdgeRegions.m*

Called by *runscript.m* to remove regions in a binary mask which are touching the edges of the image. See section 7.1.6.1.

#### 7.1.8.6 *subdivide.m*

```

function coords = subdivide(regionmask, original_img,
                            RegionSizeLowerThreshold)
%function coords = subdivide(img, original_img, RegionSizeLowerThreshold)
% Takes an image containing only one region which maybe non-rectangular
% and needs to be analyzed and divided into sub areas depending on
% whether or not objects are present.
% returns a co-ordinate list of regions ready for classification in the
% format: [ topleft_col topleft_row width height; ... ]
%
% Craig Holdaway. Last updated 23 2 04 Version 2.02

compactnessThreshold = 15;

coords = [];

%1. apply mask *****
img = im2double(original_img);
blackmasked = regionmask .* img;
whited = blackmasked + double(~regionmask);

%2. regionalize *****
%threshold:

```

```

level = graythresh(whited);
i1 = im2bw(whited, level);

%3. fill *****
%this assumes that the pollen has complete edges after thresholding.
%this is a reasonable assumption because if they weren't then the pollen
%would either be out of focus or broken, and rejectable anyhow.
i2 = imfill(~i1, 8, 'holes'); %Inversion to get black bkgnd; 8 connectivity.

%4. get region statistics *****
i3 = bwlabel(i2, 8); %8 connectivity
total_regions = max(max(i3));

for i=1:total_regions
    i4 = double(i3 == i); %select one region
    %boundary smoothing
    se = strel('square',3);
    i5 = imclose(i4, se);

    %extract properties
    temp = regionprops(i5,'Centroid');
    cent = temp.Centroid; %cent: 1st element is horz(x), 2nd ele. is
vert(y)
    area = bwarea(i5);
    iPerim = bwperim(i5);
    perim = sum(sum(double(iPerim == 1))); %pixel count

    %compactness = perimeter.perimeter / area.
    %min: circle = 12.6, square = 16, equ triangle = 20.78, 9:1 rectangle =
44    compactness = perim*perim / area;

    %if its less than the compactness thold then we believe it could be a
pollen.
    if((compactness < compactnessThreshold) && (area >
RegionSizeLowerThreshold))
        tempbounds = regionprops(i5,'BoundingBox');
        boundingbox = ceil(tempbounds.BoundingBox);
        coords = [coords ; boundingbox];
    end
end

```

#### 7.1.8.7 *removeEdgeRegions2.m*

Wrapper around the original `removeEdgeRegions.m` to provide some additional functionality.

```

function result = removeEdgeRegions2(region, RegionSizeLowerThreshold);
%function result = removeEdgeRegions2(region, RegionSizeLowerThreshold);
% Removes the edge regions from a regionalized pollen.
% Includes preprocessing then calls removeEdgeRegions.
% Arg: Region to be processed
% Returns: Region with edge regions diminished.
%
% Craig Holdaway. Last updated 24 2 04 Version 2.02

%binarize
level = graythresh(region);

```



```

i1 = im2bw(region, level);
se = strel('square',7);
%make the black bits bigger :)

i2 = double(~i1);
i3 = bwlabel(i2, 8); %8 connectivity
original_total_regions = max(max(i3));
total_regions = original_total_regions;

%dilate until (but not including) a merge occurs...
while(total_regions < original_total_regions)
    i3 = imdilate(i2, se);
    i4 = bwlabel(i3, 8); %8 connectivity
    total_regions = max(max(i4));
    if(~(total_regions < original_total_regions))
        %if no merge has occurred, allow dilate:
        i2 = i3;
    end
end
img = i2;

%%copied from removeEdgeRegions.m
[ydim,xdim] = size(img);
mask = zeros(ydim,xdim);
vertedge = ones(ydim,2);
horzedge = ones(2,xdim);
mask(:,1:2) = vertedge;
mask(:,xdim-1:xdim) = vertedge;
mask(1:2,:) = horzedge;
mask(ydim-1:ydim,:) = horzedge;
ix1 = and(mask, img);

[r,c] = find(ix1);

if(size(r)>0)
    %create black image
    [m,n] = size(img);
    iout = zeros(m,n);

    %identify regions connected to edges
    i2 = bwselect(img,c,r,8);
    i3 = bwlabel(i2,8); %8-connectivity
    total_regions = max(max(i3));
    for i=1:total_regions
        area = bwarea(double(i3 == i));
        if(area < RegionSizeLowerThreshold)
            %if this is an area to be eliminated then add it to the mask
            iout = or(iout, (i3 == i));
        end
    end
else
    iout = img;
end

i3 = imfill(~iout, 8, 'holes');

result = im2double(region) .* i3;

```

```
result = result + double(~i3);
```

#### 7.1.8.8 OnePollenClassify.m

This function performs the classification of a region within an image containing some pollen.

```
function result = OnePollenClassify(img)
% function result = OnePollenClassify(img)
% OnePollenClassify.m
% Prepares and classifies a single pollen presented to the system.
% Notes: img must be square.
%
% Craig Holdaway. Last updated 12 2 04 Version 1.99

%calculate features
featureVector = polFeatureCalculation(img);

%normalize
load diffmax46 diffmax
load collectmean46 collectionMean
normVector=(featureVector - collectionMean) ./ diffmax;

%reduce
finalVector = [normVector( 1), normVector( 2), normVector( 4), normVector(
5), normVector(18), normVector(20) ...
               normVector(24),          normVector(25),          normVector(32),
normVector(35), normVector(41), normVector(46)];

%classify
load a1 a1 %weight matrix
load a2 a2 %weight matrix
hidden1 = sig(finalVector',a1);
output = sig(hidden1,a2);

nnoutput = find(output==max(output))';

switch nnoutput
    case {1, 5, 9, 13}
        result = 'BP'
    case {2, 6, 10, 14}
        result = 'CD'
    case {3, 7, 11, 15}
        result = 'PR'
    case {4, 8, 12, 16}
        result = 'RO'
    otherwise
        error('Invalid Result');
end
```

## 7.2 Reinventing the Microscope in the Age of Digital Imaging

C.A. Holdaway and R.M. Hodgson  
Institute of Information Sciences and Technology,  
Massey University, Palmerston North, New Zealand  
c.a.holdaway@massey.ac.nz, r.m.hodgson@massey.ac.nz

### Abstract

Human requirements place many constraints on compound microscopes making them complex and expensive instruments. By replacing human eyes with an image sensor a simpler design can be used. This paper considers the design of a microscope with an image sensor which optically magnifies an object 20 times using a single lens to achieve a real magnification of over 500 times. A single lens design is analyzed but found to be inadequate so a standard microscope objective is used.

**Keywords:** microscope, pollen, image sensor

### 1 Introduction

Pollen identification is important to agriculture, horticulture and healthcare, but is time consuming because the extraction and modification of pollen is done by manual preparation and analysis [1]. Some research has been done into automating the identification of pollen [2,3], based on the use of the texture features on images of pollen [4]. While this research shows promise, it does not address the problem of image acquisition. In current techniques, pollen slides are analysed under a microscope and a trained researcher will take 2-10 hours to analyse a slide[5].

Imaging pollen introduces some application specific constraints. Pollen analysed in this study range in size from 20 $\mu$ m to 80 $\mu$ m. To be useful for texture based classification the images must contain as much useful texture information as possible. The finest texture on a pollen grain is much smaller than can be imaged using visual light, so it is necessary to settle for the image provided by magnifying the pollen to the limits of optical resolution, calculated using Rayleigh's criterion and approximated to 900nm. To obtain visible images of objects of this size a compound microscope is conventionally required.

### 2 Analysis

The design of modern compound microscopes is constrained by its intended user, that is the human being. Humans place demands on the magnification, image quality, illumination levels and packaging of a microscope. In particular our eyes, with their integral lens, are not biologically designed to view microscopic images and therefore require optics to provide magnifications in the range 400x – 1000x in

order to view pollen. To obtain magnification of this magnitude and preserve image quality requires complex optics, as simple optics designed to provide high magnification alone will exhibit severe image degradation due to lens aberrations.

A typical laboratory microscope suitable for viewing pollen has 8-15 lenses in the objective and 2-5 lenses in each eyepiece. Given that every lens added to the microscope adds expense, it is logical to seek a means of replacing our eyes with a technology that does not require such high magnification, and is more suitable for analysing and classifying microscopic objects. If the intention is to use digital image processing methods and a solid state camera to replace the human, then many lenses and features of the modern precision microscope can be dispensed with.

An image sensor, such as a CCD, is microscopic in nature. The size of an element, a pixel, on the sensor is similar in magnitude to the microscopic detail that is of interest. This substitution of a sensor for the human eye reduces the magnification required, and has two effects on the design of the optics:

1. Reduced magnification reduces the number of lenses required to enlarge the object
2. Reduced magnification reduces the number of lenses required to correct for aberrations introduced by high magnification.

Thus significant savings in optics, and therefore cost, can be made by removing from the microscope the optics that are required to meet the needs of human eyes. However this is not the limit of the possible simplification. If a microscope could be built using just a single lens it would vastly simplify the construction and so reduce the cost.

### 3 Feasibility of a Single Lens Microscope

The concept of a single lens microscope coupled to an image sensor has been implemented at low magnifications by Intel. The Intel QX3™ Toy Microscope has an integrated camera that captures images of specimens at 10×, 60× and 200× magnification. The optical magnification for the 200× magnification is performed by a single custom lens which magnifies 4x [6].

The QX3 has limitations which prevent it being used as a scientific instrument. Its highest magnification, nominally 200x, is inadequate for resolving the textural detail in pollen that is required for classification. Additionally the sensor used in image capture contains only 320×240 elements, which gives a restricted view of the scene.

In order to overcome these limitations the power of the optical system used in the QX3 would need to be increased. This would push the system into optical limitations such as aberrations and diffraction.

#### 3.1 Resolution

The fundamental optical limitation is the resolution of the optical system. Under incoherent light this is given by:

$$d_{\min} = \frac{0.61\lambda}{n \sin \alpha} \quad (1)$$

where  $\lambda$  is the wavelength of radiation,  $n$  is the refractive index and  $\alpha$  is the aperture angle of the lens [7].

To calculate this the fixed parameters of the system must be known. They are the lens focal length( $f$ ) and the sensor size( $R_{\text{img}}$ ), the latter represented as a maximum distance from the centre of the sensor to and edge, which is also the maximum extent of the lens needed for imaging. Figure 1 shows these parameters.

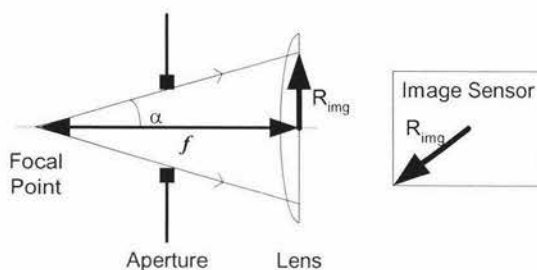


Figure 1: The parameters used for calculating  $\alpha$ .

Using the largest unfiltered wavelength,  $\lambda=700\text{nm}$ , and a refractive index,  $n = 1.664^1$ , the resolution is calculated as  $d_{\min}=2.17\mu\text{m}$ .

It is not possible to produce perfect images. Image fidelity is limited by lens aberrations and ultimately, diffraction. However if the degradation cannot be resolved by the sensor or perceived by a human then the image can for practical purposes be considered to be perfect.

#### 3.2 Aberrations

All spherical lenses produce aberrations, phenomena that degrade the image. The most obvious of these is chromatic aberration, where light of different wavelengths is refracted differently causing 'rainbow edges' in colour images, and blurring in monochromatic images. Using white light at 20 times magnification the blurring from chromatic aberrations will be in the order of  $20\mu\text{m}$ . This means that two overlapping objects less than  $40\mu\text{m}$  will blend with each other.

Spherical aberration is caused by different regions of the lens focusing light at different distance from the lens and causes general blurring to an image. Spherical aberration can be calculated using ray-tracing techniques such as those explained in [8]. There are two key factors that affect the spherical aberration:

1. The aperture size limits the maximum ray divergence from the object, limiting the divergence of meridional rays. Therefore the spherical aberration is reduced as the aperture is closed.
2. The refractive index of the lens determines the severity of the refraction at the surfaces. A smaller refractive index causes less divergence and therefore less spherical aberration.

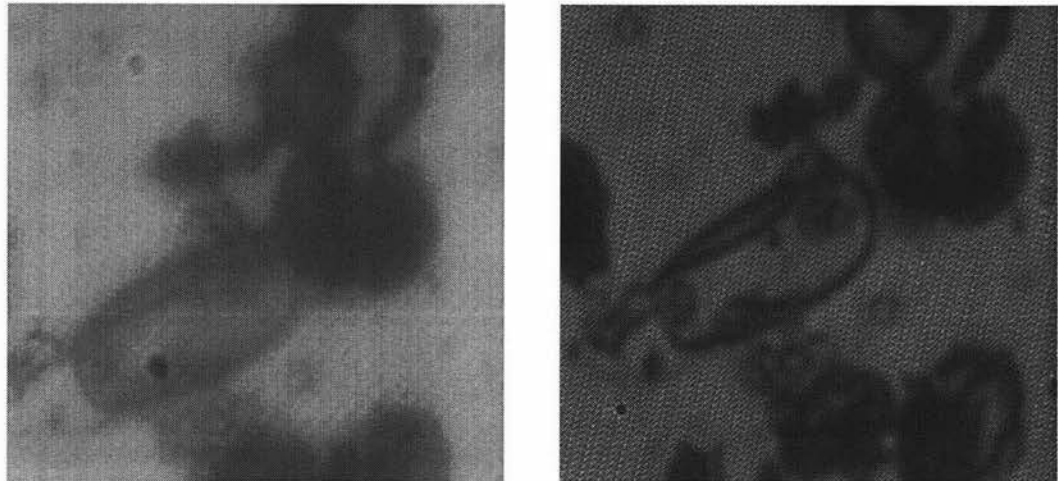
#### 3.3 Diffraction

Diffraction also limits the resolution. An aperture has been introduced into the system to control the illumination and spherical aberration. As the aperture is a limiting edge it is a barrier which light diffracts around. Using a single-dimension simplification, it can be shown that the spread of the central maximum of the diffraction emanating from a point within a slit has the following formula:

$$W = \frac{2L\lambda}{b} \quad (2)$$

where  $W$  is the width of the central maxima,  $L$  is the distance from the slit to the image,  $b$  is the slit width, and  $\lambda$  is the wavelength of the light[9].

<sup>1</sup> The lens with these parameters is A45-097 from Edmund Optics [11].



**Figure 2:** Effect of applying a narrowband filter to the light source.

### 3.4 Correcting Aberrations

Correction of chromatic aberration typically requires an achromatic doublet, which introduces a negative lens with a high refractive index. An alternative solution is to limit the wavelength of the source. During early experimentation a filter was applied consisting of a Near-IR Cut Filter on the camera transmitting  $\lambda < 700\text{nm}$ , and a red filter on the source transmitting  $\lambda > 630\text{nm}$ . Thus the illumination has a bandwidth of  $70\text{nm}$ , compared to the bandwidth of white light at  $\sim 320\text{nm}$ . The effect is to improve the detail in the image as shown in Figure 2.

This is an acceptable limitation for our application, and many image processing applications, where colour information is not used.

Correcting for spherical aberration requires that the aperture diameter be decreased. However it can be seen from equation 2 that decreasing the aperture (parameter  $b$ ) increases diffraction. Thus we have a trade-off between aberration and diffraction.

Applying equation 2 shows that an aperture greater than  $21\text{mm}$  is required to reduce the effect of diffraction to the desired value of  $2\mu\text{m}$ . However this increase in aperture size increases the quantity of non-paraxial rays in the system and increases the magnitude of the aberrations.

In order to correct for spherical aberration the diameter of the point spread function (p.s.f.) should be at most  $2.2\mu\text{m}$ . To achieve this an aperture of diameter  $0.061\text{mm}$  is required. At this aperture diameter the effect of diffraction would be a p.s.f. with diameter  $12\text{mm}$  or about the half the area of the imaging sensor. Additionally an aperture of this size would block out 99.9% of the image.

### 3.5 Conclusion

This trade-off between aberrations and diffraction in the optical system prevents achievement of the optimal resolution. Although the values here are for the case of a lens with  $f=25$ , a lens where the diameter of both the spherical aberration and beam spread is smaller than  $2\mu\text{m}$  cannot be found by applying the ray tracing and beam spread equations to lenses with both shorter and longer focal lengths.

Therefore, a single lens microscope magnifying twenty times cannot provide the resolution required for imaging pollen.

## 4 Design of a Computer Microscope

### 4.1 Design Alternatives

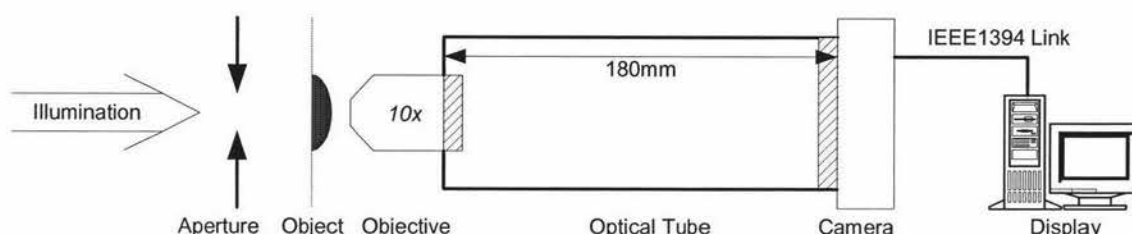
There are two likely alternatives for the basic design of a computer microscope.

The first option, custom optics, allows greater design flexibility. Triplets, such as Cooke's Triplet[10], can adequately correct for all primary aberrations and chromatic aberration. However they require considerable optical and mechanical design.

The second option, a standard finite achromatic microscope objective, is less flexible than custom optics, but has known design parameters, is mechanically housed, and contains corrective optics for all primary aberrations. Importantly, the cost of a standard objective and the cost of three lenses plus housing is approximately equal[11].

As this is an integration application the standard microscope objective is the prudent and expedient option and has been selected.





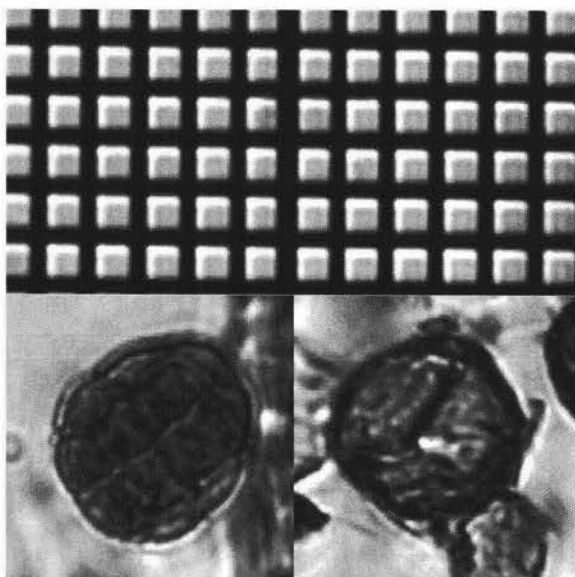
**Figure 3:** Design of a Computer Microscope to optically magnify 12×

## 4.2 Selected Design

The selected design for a computer microscope, shown in Figure 3, is vastly simpler than a conventional compound microscope. Most importantly the optical magnification has been greatly reduced. At lower magnifications illumination is not as critical as it is for higher magnifications. This removes the need for complex and expensive condenser optics and illumination train and allows a cheap source of illumination to be used, such as an incandescent light bulb.

This computer microscope is more flexible than a conventional compound microscope as it is not bulky or fixed in place, making it suitable for field work.

This computer microscope stands apart from existing entry level computer microscopes such as the QX3 by providing significantly higher magnification.



**Figure 4:** Image of a calibration grid and two pollen captured using the computer microscope at 12x optical magnification

## 5 Results

A standard 10× achromatic objective lens was used to acquire the images. By using different optical tube lengths magnifications in the range 10-22x we obtained. It was found that above about 12× increasing the magnification did not increase the resolution. Furthermore, the 12× images, displayed in Figure 4, contain pollen of about the right size and of sufficient detail to be classified using the existing classification algorithm.

The contrast of the images captured could be improved by correction and, expensive, illumination. However this degradation is not critical and can be improved in software by using contrast expansion algorithms.

## 6 Conclusion

If an optical microscope is not designed for direct human viewing, then the need for complicated optics is eliminated. This makes it possible to construct a simple microscope using only a standard microscope objective and a digital image sensor. As a consequence of the direct use of a solid state sensor having microscopic elements to image an object, the resulting microscope has lower optical magnification than a conventional microscope. Additionally the computer microscope does not need expensive illumination systems and can be positioned flexibly.

## 7 References

- [1] E. C. Stillman and J. R. Flenley, "The Needs and Prospects for Automation in Palynology", *Quaternary Science Reviews*, Vol 15, pp1-5, 1996
- [2] P. Li and J. R. Flenley, "Pollen Texture Identificaiton using Neural Networks", *Grana* 38, pp59-64, 1999.
- [3] J.R.Flenley, P. Li, L. K. Empson, "Identification of 13 Pollen Types by Neural Network Analysis of Texture Data Only", in *Proceedings of Image and Vision Computing New Zealand*, 1999, pp295-298.
- [4] Y. Zhang, D. Fountain, J. Flenley, R. Hodgson, S. Gunetileke "Pollen Patterns Recognition using

Gabor Transforms and Digital Moments”, under review.

- [5] J. R. Flenley, “The problem of Pollen Recognition”, in M. B. Clowes and J. P. Penny, *Problems in Picture Interpretation*, pp141-145. CSIRO, Canberra, 1968.
- [6] L. Jelinek, G. Peters, J. Okuley, S. McGowan, “Dissection of the Intel Play QX3 Computer Microscope”, *Intel Technology Journal Q4*, 2001, pp1-10.
- [7] E. M. Slayter. H. S. Slayter, “9.4 Theories of ultimate resolving power” in *Light and Electron Microscopy*, Melbourne, Australia. Cambridge University Press, 1992, pp.121-126
- [8] W. J. Smith, “Optical Computation” in *Modern Optical Engineering*, USA. McGraw-Hill, 1966. pp.247-253
- [9] F. L. Pedrotti. L. S. Pedrotti, “16-2 Beam Spreading” in *Introduction to Optics*, 2nd Ed. New Jersey: Prentice-Hall, 1993, pp329-330.
- [10] W. J. Smith, “The Design of Optical Systems” in *Modern Optical Engineering*, USA. McGraw-Hill, 1966. pp.340-347
- [11] Edmund Optics, *Optics and Optical Instruments Catalog*, New Jersey. 2002. Online copy available at [www.edmundoptics.com](http://www.edmundoptics.com)

## 7.3 An Introduction to Wavefront Coding

Craig Holdaway  
Institute of Information Sciences and Technology  
Massey University

### Introduction

Wavefront coding uses aspheric optical elements and digital signal processing to enhance the performance of imaging systems. Wavefront coding has been applied to imaging systems to extend depth of field and to control aberrations. These applications have demonstrated at least an equivalent image quality as a more expensive traditional optical system, with 4-5x greater field of view without changing aperture sizes in the system[1]-[4].

### Background

Depth of field has traditionally been extended by altering the exit pupil definitions,. Usually by stopping down the aperture until the desired focal depth has been reached. However this creates some problems[3]:

- The illumination through the system is reduced, which in turn requires an increase in the exposure and increases the likelihood that the object will move during imaging.
- A smaller aperture loses the high frequency information encoded in the more widely diffracted rays, blurring the image

Wavefront coding introduces a phase plate into the aperture stop of the optical system. The phase plate is designed to generate a known blur that creates invariance to many optical aberrations including: spherical aberration; astigmatism; field curvature; chromatic aberration and; defocus.[1]

The blurred intermediate image captured by the image sensor is processed by digital signal processing that removes the blur and changes the phase of the spatial frequencies. The system as a whole is illustrated in Figure 1.

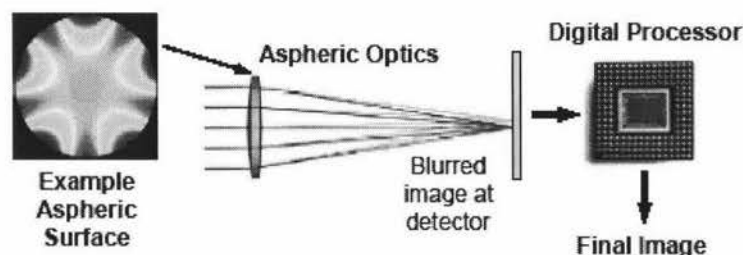


Figure 1 Wavefront coded imaging system[1]

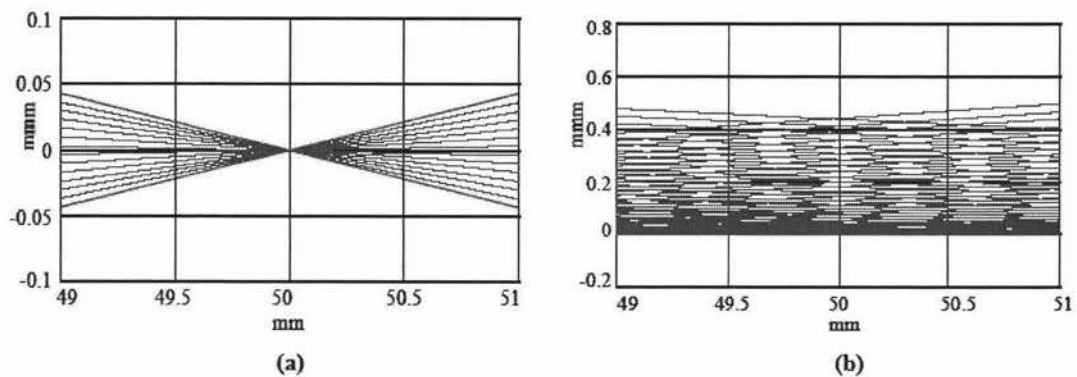
Current design methods for optical systems using wavefront coding involve careful mathematical design of the phase plate to achieve aberration invariance, then selection of a signal processing algorithm to restore the original image. The design approach treats the object as a 2-D signal source and the mathematics is generally done in the spatial frequency domain, by manipulating the OTF of the subsystems, as the optical wavefronts can be considered to be convolved with the optical elements.

### Optics

Figure 2a shows a ray diagram for a traditional imaging system. The image plane must be located precisely, at the 50mm mark, for the image to be in focus. Figure 2b shows a ray diagram for the same system after a cubic phase plate is inserted at the aperture stop. Moving the image plane within the boundaries of Figure 2b does not affect the resulting image greatly. Looking at this conversely, for a fixed image plane, in a wavefront coded optical system, many object planes will generate a coded

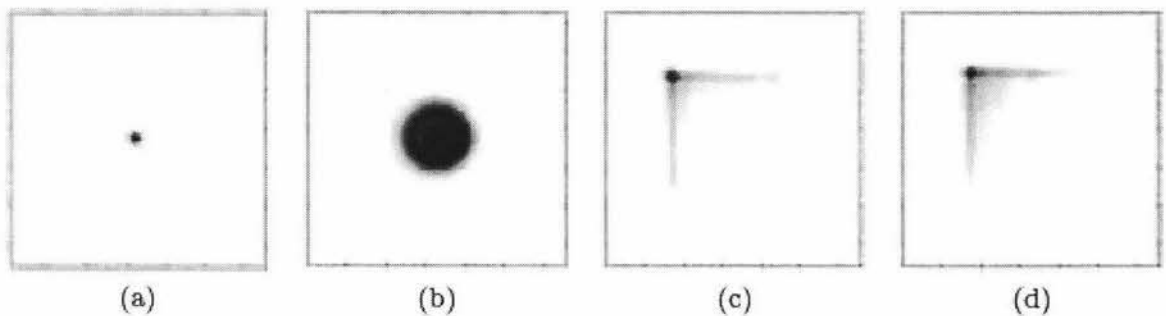


image that can be digitally processed from a single image plane. Thus the depth of field of the wavefront coded system is much greater than the traditional system.



**Figure 2: Ray-based explanation of Wavefront Coding. [4]**

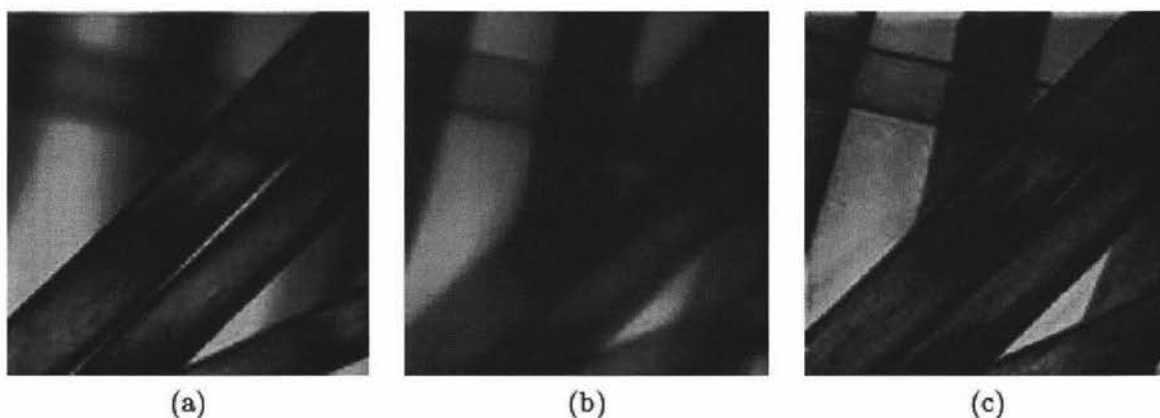
The PSF generated by an optical system with a cubic phase inserted is shown in Figure 3. The invariance to defocus can be seen in Figure 3, where the effect of ten waves of defocus on the standard PSF (Figure 3a) generates significant blur (Figure 3b), whereas the effect of ten waves of defocus on the PSF of a wavefront coded system (Figure 3c) is minimal (Figure 3d).



**Figure 3 Images of standard PSFs with (a) zero waves of defocus, (b) ten waves of defocus, next to images of cubic phase play system PSFs with (c) zero waves defocus and (d) ten waves of defocus[2].**

## Results

The improvement in depth of field is visible in Figure 4, where the background hairs which are blurred in Figure 4a) are sharp in Figure 4c). There is also greater detail of the contents within the hairs.



**Figure 4** Images of human hair with average thickness  $65\mu\text{m}$  taken using (a) a standard optical system, (b) the intermediate cubic phase plate system image, and (c) the processed cubic phase plate system image[2].

### Application to Microscopy

Applying wavefront coding to microscopy provides higher quality images for low power microscopes due to the invariance to degrading primary aberrations. This application also reduces the time needed for some laboratory work where under a traditional microscope multiple image slices had to be taken to get an image with the depth that is achieved by a microscope using wavefront coding.

However application of wavefront coding to high magnification microscopes is more difficult due to the precision needed to located the apertures, high system resolution, and the optimization of digital processing algorithms.[2]

### Conclusion

Wavefront coding is a cheaper means of improving the performance of an optical system. By carefully designing the system parameters, the effect of aberrations can be removed and the depth of field significantly increased compared to conventional optical systems of the same power and numerical aperture.

### Glossary

PSF – Point Spread Function. The spatial domain representation of the effect of the optical system on the input.

OTF – Optical Transfer Function. The spatial frequency representation of the effect of the optical system on the Fourier transform of the input.

### References

- [1] K. Kubala, E. Dowski, W. T. Cathey, "Reducing complexity in computational imaging systems," *Optics Express*, vol. 11, no. 18, pp. 2102-2108, 8 Sept 2003.
- [2] S. C. Tucker, W. T. Cathey, E. R. Dowski, "Extended depth of field and aberration control for inexpensive digital microscope systems," *Optics Express*, vol. 4, no. 11, pp. 467-474, 24 May 1999.
- [3] S. Bradburn, W. T. Cathey, E. R. Dowski, "Realizations of Focus Invariance in Optical/Digital Systems with Wavefront Coding," *Applied Optics*, vol. 36, December 1997.
- [4] E. R. Dowski, G. E. Johnson, "Wavefront Coding: A modern method of achieving high performance and/or low cost imaging systems," *Proc. SPIE*, August, 1999.

## 7.4 Bibliography

M. Born, E. Wolf, "5.3 The primary (Seidel) aberrations" in *Principles of Optics*, 4<sup>th</sup> Ed., Pergamon Press. 1970.

K. R. Castleman, *Digital Image Processing*. Englewood Cliffs, N.J.: Prentice Hall, c1996.

T. N. Cornsweet, *Visual Perception*, New York: Academic Press Inc., 1970. pp.7-9.

M. W. Davidson, M. Abramowitz, Olympus America, Florida State University, *Optical Microscopy Primer*. [www.microscopy.fsu.edu/primer/](http://www.microscopy.fsu.edu/primer/), 2004.

R. C. Gonzalez, R. E. Woods, *Digital Image Processing*, 2<sup>nd</sup> ed. Prentice-Hall: New Jersey, 2001.

R. B. Knox, *Pollen and Allergy*, Southhampton: Edward Arnold Publishers, 1979.

F. Pedrotti, L. Pedrotti, *Introduction to Optics*. Englewood Cliffs: N.J. Prentice-Hall International, 1993

R. S. Pressman, *Software Engineering, A Practitioner's Approach*, 5<sup>th</sup> Ed. McGraw-Hill, 2001. p.488

J. C. Russ, *The image processing handbook*, 2<sup>nd</sup> Ed. Boca Raton, Florida: CRC Press, c1995

E. M. Slayter, H. S. Slayter, *Light and Electron Microscopy*. Cambridge University Press, 1992, p.131-148.

Warren J. Smith, *Modern Optical Engineering*. McGraw-Hill, 1966.

B. Walker, *Optical Engineering Fundamentals*. McGraw-Hill Inc., 1995.

T. E. Weier, M. G. Barbour, C. R. Stocking, T. L. Rost, *Botany*. John Wiley & Sons: Singapore, 1982.

H. D. Young, R. A. Freedman, *University Physics*. Addison-Wesley: Reading, Mass. 1996.

Y. Zhang, *Pollen Discrimination Using Image Analysis*, Postdoctoral Research Report, Massey University, New Zealand, 2003