A Procedure for Designing and Assessing the Performance of Image Processing Systems

Samuel Gerard Bailey
Sira/UCL PTP
Image Processing Group,
Department of Physics and Astronomy,
University College London
and
Sira Technology Centre

A thesis submitted for the Degree of
Doctor of Philosophy
University of London
2000
Abstract

The field of image processing (IP) currently lags behind many other fields in science and engineering in the development of techniques for predicting and assessing system performance. This thesis describes a technique for assisting developers and users of IP systems. It presents a methodology for the design and the performance prediction of those systems in different imaging conditions.

The thesis surveys various performance analysis techniques which have been developed to analyse IP system performance, namely benchmarking, performance evaluation and performance characterisation. It outlines the differences, as well as the advantages and short-comings of each technique. The thesis then presents a new methodology to guide system designers in gathering the appropriate data about imaging conditions, designing the IP system, and predicting and assessing its performance.

The methodology operates by guiding the developer through the following stages: Firstly appropriate parameters are selected to describe the imaging conditions and the final performance metrics. These are narrowed down until only the most important factors remain. The nature of these parameters is then used to determine the best approach to performance analysis, either analytical, empirical, or a combination of the two. An example algorithm is then chosen which could be used to perform the IP task. This algorithm is then modularised, or broken down into its constituent components. These modules are then analysed one by one to determine which imaging parameters affect which module, and what internal quality propagation parameters can be used to measure the effect that the performance of each has on the other modules. Transfer functions are then derived which relate how incoming parameters effect outgoing metrics for each module. Finally, the performance of the different modules is combined, together with a distribution of the operating conditions to produce a final performance measure for the system.

The effectiveness of the methodology is demonstrated by applying it to four industrial image-processing systems: visual tracking of batches of steel, automatic identification of batches of steel, lens aberration determination in a transmission
electron microscope and fuel drum location for automatic materials handling. In each case an example algorithm is chosen to perform the task, and its performance is predicted under the operating conditions it is likely to encounter. The methodology’s applicability to two further tasks is also shown, and conclusions and recommendations for further work are discussed.
Acknowledgements

Many people made this thesis possible, however the following deserve a special note of appreciation. My primary supervisor Dr. Michael Forshaw of the Image Processing Group (IPG), University College London, gave an enormous amount of support, and was a constant stream of encouragement and excellent ideas. His assistance was particularly appreciated in helping develop some of the TEM models used. My secondary supervisor, Dr. Mark Hodgetts of Sira Ltd, also provided a large amount of encouragement and useful input. I would also like to thank Dr. Angus Kirkland of JEOL (UK) Ltd., Ken Ridgeway of British Steel Technology Centre and Dr. David Case of BNFL Engineering Ltd. Each provided me with extremely useful examples of industrial image processing systems, as well as data, images, ideas and support. Thanks are due too to Dr. Jonathan Barker of the IPG for some of his elegant mathematical insights, and his assistance in allowing me to analyse his Cancer Diagnosis system. They are also due to Stewart Young of the IPG for providing a further example to demonstrate my methodology, and for acting as a sounding board for many of my ideas. I also appreciate the help of the other members of the IPG, particularly Dr. David Crawley for his valuable assistance with the finer points of computing, and Dr. Terry Fountain, who also provided ideas and feedback.
# Table of Contents

ABSTRACT ......................................................................................................................................................... 2

ACKNOWLEDGEMENTS ........................................................................................................................................ 4

TABLE OF CONTENTS ...................................................................................................................................... 5

LIST OF FIGURES ........................................................................................................................................... 10

LIST OF TABLES .............................................................................................................................................. 14

CHAPTER 1: INTRODUCTION .......................................................................................................................... 15
  1.1 MOTIVATION – THE NEED FOR PERFORMANCE DATA ................................................................. 15
  1.2 DEVELOPING PERFORMANCE PREDICTION TECHNIQUES .................................................... 16
  1.3 SCOPE OF THIS RESEARCH ............................................................................................................. 17
  1.4 STRUCTURE OF THIS THESIS ........................................................................................................ 18

CHAPTER 2: BACKGROUND ............................................................................................................................. 21
  2.1 PERFORMANCE MEASUREMENT IN OTHER ENGINEERING DISCIPLINES ............................. 21
  2.2 DIFFERENT TECHNIQUES FOR MEASURING PERFORMANCE .................................................. 22
    2.2.1 Performance Characterisation ................................................................................................. 22
    2.2.2 Performance Evaluation ............................................................................................................ 25
    2.2.3 Benchmarking ............................................................................................................................ 25
    2.2.4 Modularisation ......................................................................................................................... 27
  2.3 RELATED WORK ...................................................................................................................................... 27
    2.3.1 Standardisation ............................................................................................................................ 28
    2.3.2 Standard Frameworks ............................................................................................................... 28
  2.4 BACKGROUND ON TECHNIQUES ..................................................................................................... 30
    2.4.1 Template Matching .................................................................................................................... 31
    2.4.2 Tracking ....................................................................................................................................... 32
    2.4.3 Transmission Electron Microscopy Lens Aberration Analysis Techniques ................................ 34
    2.4.4 Error Correcting Codes ............................................................................................................. 44
    2.4.5 Types of Error-Correcting Codes ............................................................................................. 48
  2.5 SUMMARY ............................................................................................................................................. 49

CHAPTER 3: METHODOLOGY .......................................................................................................................... 50
  3.1 TYPICAL INDUSTRIAL SYSTEM DEVELOPMENT .................................................................... 50
  3.2 PERFORMANCE ANALYSIS METHODOLOGY ....................................................................... 51
  3.3 INTRODUCTION ................................................................................................................................. 53
    3.3.1 The Effect of Operating Conditions on Performance .............................................................. 53
CHAPTER 4: LADLE TRACKING ........................................................................................................... 69

4.1 OUTLINE OF PROBLEM ........................................................................................................... 69
   4.1.1 Image Processing System ................................................................................................. 70
   4.1.2 Maximum Finder .............................................................................................................. 72

4.2 PROBLEM ANALYSIS ............................................................................................................. 73
   4.2.1 Parameters and Ranking ................................................................................................ 73
   4.2.2 Performance Metric ......................................................................................................... 74
   4.2.3 Classification ................................................................................................................... 74

4.3 SYSTEM ANALYSIS ................................................................................................................ 75
   4.3.1 Interaction of Parameters and Quality Measures ............................................................. 75
   4.3.2 Simplification of Analysis ............................................................................................... 76

4.4 ANALYSIS OF MODULES ..................................................................................................... 77
   4.4.1 Modules 1 and 2: Edge Filter and Template Matching .................................................... 77
   4.4.2 Module 3: Maximum Finder ............................................................................................ 79

4.5 ESTIMATE OF PERFORMANCE .......................................................................................... 82

CHAPTER 5: LADLE IDENTIFICATION ............................................................................................. 84

5.1 BACKGROUND ......................................................................................................................... 84
B.1.1 The Problem ............................................................................................................................................................................... 210
B.2 Performance Analysis ...................................................................................................................................................................... 214
  B.2.1 Problem Level ........................................................................................................................................................................ 214
  B.2.2 System Level ........................................................................................................................................................................... 214
B.3 Intruder Detection ........................................................................................................................................................................... 216
  B.3.1 The Problem ........................................................................................................................................................................ 216
  B.3.2 Description of the IP .............................................................................................................................................................. 217
B.4 Performance Analysis ...................................................................................................................................................................... 219
  B.4.1 Problem Level ........................................................................................................................................................................ 219
B.5 Summary ...................................................................................................................................................................................... 221

BIBLIOGRAPHY .................................................................................................................................................................................. 222

PUBLICATIONS .................................................................................................................................................................................. INSIDE BACK COVER

Goal Orientated Performance Evaluation, workshop on performance characterisation, 1st ICVS, Las Palmas de Gran Canaria, Jan 1999

A Methodology for Designing and Assessing the Performance of Image Processing Systems, submitted to Computer Vision and Image Understanding, Special Issue on Performance Evaluation
List of Figures

Figure 2-1 A diffractogram from a transmission electron microscope. 36
Figure 2-2 Use of 'sector averaging' technique on non-circular diffractograms 37
Figure 2-3 Synthetic diffractogram with defocus only 40
Figure 2-4 Spectrum of figure 2-3 40
Figure 2-5 Cross-section through spectrum 40
Figure 2-6 Synthetic diffractogram with defocus and astigmatism 40
Figure 2-7 Spectrum of figure 2-6 40
Figure 2-8 Cross-section through spectrum 40
Figure 2-9 Synthetic diffractogram with spherical aberration, defocus and astigmatism 41
Figure 2-10 Spectrum of figure 2-9 41
Figure 2-11 Spectrum after multiplication by $\exp(-\frac{1}{2}i\pi\lambda C_{1}r^{4})$ 41
Figure 2-12 Cross section through spectrum before and after multiplication by $\exp(-\frac{1}{2}i\pi\lambda C_{1}r^{4})$ 41
Figure 2-13 Diffractogram handfitted to real data 41
Figure 2-14 Spectrum of figure 2-13 41
Figure 2-15 Cross section through spectrum 42
Figure 2-16 Illustration of the technique for finding the angle of two-fold astigmatism for a diffractogram 44
Figure 2-17 Illustration of the concept of distance in code of length n=3. 45
Figure 2-18 Illustration of the concept of distance in code of length n=3. 46
Figure 2-19 Some of the commercially available visual identification codes. 47
Figure 3-1 Block diagram showing the steps in the formation and analysis of an image [37] 51
Figure 3-2 The parallel performance analysis of an algorithm 55
Figure 3-3 Block diagram of a simple tracking algorithm for use in tracking steel ladles. 60
Figure 3-4 Interaction of the parameters with the tracking system 61
Figure 3-5 The final performance block diagram. 63
Figure 3-6 A single module in the system 65
Figure 4-1 Input video image of ladle being transported by crane around Llanwern Steelworks. 70
Figure 4-2 Block diagram of a simple tracking algorithm for use in tracking steel ladles. 71
Figure 4-3 The final performance block diagram. These are the full set of factors which will be used for the performance analysis. 76
Figure 4-4 Simplified block diagram of the ladle tracking system. 76
Figure 4-5 Module 1: Combined edge filter and template matcher. 77
Figure 4-6 Edge filtered image $\hat{f}(m,n)$ produced using the Sobel operator. 78
Figure 4-7 Correlation surface generated by correlating the ladle template with edge filtered image 78
Figure 4-8 Graph of correlation signal depth versus ladle edge contrast. 79
Figure 4-9 Module 3: The maximum finder. 80
Figure 4-10 Probability of error, \( G(D) \), in any one matching operation as a function of correlation depth, \( D \) ____________________________________________________________________________________ 81

Figure 4-11 Frequency distribution of ladle edge strength, \( P(\alpha) \). __________________________ 82

Figure 5-1 Bimodal distribution of the pixel intensity from the two-tone pattern image ____________ 86

Figure 5-2 Modules of the ladle identification system _________________________________________ 90

Figure 5-3 The modules of the ladle identification system ______________________________________ 93

Figure 5-4 Information stream and performance parameters for the correlator module __________ 94

Figure 5-5 Cross section through template \( t(x) \), and scene \( s(x) \) with template displaced by \( \Delta x \) ______ 97

Figure 5-6 Information and performance parameters for the maximum finder module. ____________ 98

Figure 5-7 Cross section through the minimum in the correlation surface showing the effect of
discretisation when the minimum is centred on a pixel position ______________________________ 100

Figure 5-8 Discretised correlation minima not centred on a pixel position _________________________ 101

Figure 5-9 Interaction of parameters with the thresholder module _______________________________ 102

Figure 5-10 Trimodal distribution of pixel intensity when a region of the pattern is obscured. ____ 102

Figure 5-11 Parameter interaction with the bit-classifier module. ________________________________ 104

Figure 5-12 Illustration of the difference between the known region size and the actual region size, given
an uncertainty in the region position of \( \delta x \). ____________________________________________ 105

Figure 5-13 Parameter interaction with the decoder module ______________________________________ 106

Figure 5-14 Graph showing the maximum error correction rate versus code length for the codes encoding
5 information bits. ______________________________________________________________________ 107

Figure 5-15 Numeric code. Circle is used for location and digit indicates ladle identity. __________ 107

Figure 5-16 Binary circle code. Five concentric circles encode 5 bit binary ________________________ 108

Figure 5-17 As Figure 5-16 but with the bit represented by the inner ring on the top half of the pattern
switched with the outer ring, and so forth. ___________________________________________________ 108

Figure 5-18 Vertical bar code using a \( \text{BCH} \) code with \( n = 15 \) stripes, maximum of \( t=3 \) errors. ___ 108

Figure 5-19 Bullseye code. Centre bullseye and outer ring provide location information. Teeth in outer
ring encode coded version of the ladle number using a \( \text{BCH} \) code. ____________________________ 109

Figure 5-20 Block array code using a \( \text{BCH} \) code with \( n = 15 \) blocks, maximum of \( t=3 \) errors. _____ 109

Figure 5-21 The empirical correlation scores as the numerals 1 to 20 are matched against themselves
with zero noise. White squares indicate a correlation score of one, black of zero. _______________ 110

Figure 5-22 The empirical correlation scores as the numerals 1 to 20 are matched against themselves
with added noise (\( \text{SNR} = 1.4 \)). _________________________________________________________ 111

Figure 5-23 Image of ladle taken from steelworks with toothed bullseye code superimposed on the image.
_______________________________________________________________________________________ 113

Figure 5-24 Values of the various error probabilities at the different stages of processing as a function of
contrast/noise ratio ________________________________________________________________________ 115

Figure 5-25 Bullseye code with a contrast/noise ratio of 1.6 _________________________________ 116

Figure 5-26 Bullseye code appearing at a contrast to noise ratio of 0.4. __________________________ 116

Figure 5-27 Error probabilities as a function of obscuration, \( \rho \) _______________________________ 117
Figure 6-1 Image formation in a lens system: wave aberration and image aberration.  
Figure 6-2 The transmission electron micrograph of an amorphous carbon film with a crystalline inclusion in the middle.  
Figure 6-3 A set of five beam-tilt diffractograms, each obtained by Fourier transforming an image of the same amorphous carbon film specimen shown in Figure 1.  
Figure 6-4 Diffractogram fit. The top right-hand half shows the original image; the bottom half shows a hand-fitted theoretical diffractogram.  
Figure 6-5 Block diagram of the lens aberration determination system.  
Figure 6-6 Parameter interaction with the correlator module.  
Figure 6-7 Correlation score $C(\theta)$ versus rotation angle $\theta$ of diffractogram before folding.  
Figure 6-8 The three-component model of the synthetic diffractogram pattern.  
Figure 6-9 Predicted ratio of correlation value at peak ($C_{\text{peak}}$) to mean correlation value away from peak or minima ($C_{\text{mean}}$).  
Figure 6-10 Predicted ratio of the mean correlation value away from peaks or minima ($C_{\text{mean}}$) to the expected correlation value at the minima ($C_{\text{min}}$).  
Figure 6-11 By fitting a quadratic curve to the minimum of the correlation function, an approximate estimate of the apparent primary astigmatism coefficient $A_1$ can be obtained using eq(6-44).  
Figure 6-12 Illustration of the effect of noise in uncertainty in the correlation minima position.  
Figure 6-13 The parameter interaction with the correlator and minimum module.  
Figure 6-14 Actual correlation minimum for the diffractogram shown in Figure 6-4.  
Figure 6-15 Cross section through the minimum shown in Figure 6-14.  
Figure 6-16 Contour plot of the correlation function $C(A_1, C_1)$ over a wide range of $A_1$ and $C_1$.  
Figure 6-17 Variation in enclosed energy as a function of the radius of integration.  
Figure 7-1 Bin layout in UO$_3$ storage facility.  
Figure 7-2 SGV with test drums.  
Figure 7-3 UO$_3$ Drums.  
Figure 7-4 A block diagram of the image processing system.  
Figure 7-5 Simplified block diagram of the image processing system.  
Figure 7-6 Parameter propagation for the drum location system.  
Figure 7-7 Interaction of parameters with the thresholder module.  
Figure 7-8 The effect of noise on the probability of a pixel being above the threshold.  
Figure 7-9 Interaction of parameters with the linker module.  
Figure 7-10 Graph showing the probability of a pixel having a grey-level above the threshold, in the region where the illumination gradient intersects the threshold.  
Figure 7-11 Probability function of a pixel being above the threshold, $1-m(x,y)$ (dotted line) and corresponding probability of being linked, $p(x,y)$ (continuous line).  
Figure 7-12 Parameter interactions for the centroid finding algorithm.  
Figure 7-13 Original Image.  
Figure 7-14 Thresholded version of original image, $A_{\text{min}}/A = 95\%$.  

12
Figure 7-16 Above image after linking. The non-connected pixels have been removed. 195
Figure 7-17 Illumination surface 195
Figure 7-18 Predicted values of \( m(x,y)+n(x,y) \) corresponding to the thresholded image. 195
Figure 7-19 Predicted values of \( p(x,y)+q(x,y) \), scaled as above. 195
Figure 7-20 Graph of the actual positioning error (continuous line) and predicted positioning error
(broken line) in locating the drum cap as a function of the uncertainty in apparent cap size, \( A_{\text{true}}/A \). 196
Figure A-1 Error correcting Bullseye code used in chapter 5. 209
Figure B-1 An example of a specimen taken from an ovarian tumour. 211
Figure B-2 Voronoi diagram generated from cell nuclei (from [103]). 212
Figure B-3 Quadratic boundary fit to a segmented region of epithelium (from [103]). 213
Figure B-4 Diagram showing complete information streams and performance parameters for the ovarian
cancer detection system. 215
Figure B-5 Typical image from the intruder detection system showing an intruder (centre) entering the
sterile zone between the two fences. [Image courtesy of PSDB] 217
Figure B-6 Information stream and suggested parameters for the intruder detection system 220
List of Tables

Table 4-1 The parameters affecting the performance of the ladle tracking algorithm....................... 73
Table 5-1 Relative ranking of phenomena which might affect the performance of the proposed image processing system.......................................................... 88
Table 5-2 Modelling the parameters which affect the performance of the system, and the variables used to describe them................................................................. 91
Table 5-3 Comparison of the parameters for the various code designs........................................... 112
Table 5-4 Estimates of parameter values for the operating conditions in the steelworks .................... 114
Table 6-1 Parameters affecting the performance of the lens aberration determination system.......... 129
Table 7-1 Parameters affecting the performance of the IP system on the SGV.................................. 177
Table 7-2 The parameters describing the fit for each of the 12 illumination conditions tested.............. 190
Table A-1 Length, n, number of information bits, k, and distance, d of some optimum codes [78]...... 206
Table B-1 Statistics generated by the boundary detection algorithm (from [103]); .............................. 213
Table B-2 Parameters affecting the performance of the ovarian cancer detection system.................. 214
Table B-3 Parameters affecting the performance of the intruder detection system.......................... 219
Chapter 1: Introduction

The field of image processing (IP) systems currently lacks a formalised structure, or methodology, for developing and assessing the performance of such systems. This often leads to a somewhat ad hoc approach to development, with systems being built and then tested, without their likely performance being calculated beforehand. This thesis describes a novel methodology which has been developed to assist engineers in both the development of IP systems and the prediction and characterisation of their performance.

1.1 Motivation – The Need for Performance Data

One of the key issues for engineers working in the development of IP and computer vision as an engineering discipline, is the relative lack of work being done to measure the performance of image processing algorithms and systems, and the consequent lack of data. For the engineers involved, this makes the design of systems far harder [1, 2] and diagnostics more difficult [3]. This often results in a less systematic approach to development and testing, with less reliable systems being built. In turn, IP sometimes suffers from a problem of being seen as something of an immature technology and may consequently be underused. Producing a way to measure system performance or reliability will have the following main benefits:

Aiding Algorithm Development

By enabling researchers to measure the performance of their algorithm relative to existing techniques, a developer can quantify any improvements that each alteration causes and can more readily arrive at a better solution. He or she can also compare the effects of combining different algorithm components and using different values for tuning parameters.

Predicting System Suitability

If a system performance for a given task can be predicted from previously acquired data, its suitability for the task can be evaluated and testing and development time reduced.

Algorithm Selection
Knowledge of the performance of different algorithms can aid the selection of the most appropriate technique for a given problem.

**System Optimisation**

If an envelope describing the performance of a system can be created, then this can be used as part of the input to a system optimisation technique. This will ultimately enable the optimum operating point for a system to be determined given the constraints of the task.

It is thus imperative that image processing starts to develop the sort of tools and information for performance measurement that are taken for granted in other disciplines of engineering.

### 1.2 Developing Performance Prediction Techniques

Predicting and characterising the performance of IP systems is difficult, mainly for the following reasons (some from [4]):

- Performance evaluation is task dependent. The overall system performance is a function of both the effectiveness of the algorithm, and the conditions under which it is operating. It is usually necessary to decouple these two factors. This enables the performance of a given system to continue to be predicted even when the operating conditions change, and brings universal performance measurement a step closer.
- Different tasks require different performance measures. The metrics used to characterise the performance of a tracking algorithm will differ from those which are used to measure the performance of an optical character recognition algorithm, and must be selected appropriately.
- The operating conditions must be characterised. Describing complex imaging conditions quantitatively is difficult.
- Vision is often only one component of a larger system. Other non-imaging factors may affect the overall performance, which must be taken into account.
- Vision is complex. The vision system often consists of several component algorithms which are combined to solve a vision task.
- The models used to describe images and imaging systems are sometimes incorrect. Prediction must make do with sub-optimal models.
Different performance measures mean that different systems cannot be compared directly.

Algorithms are often developed without an accompanying theory, which makes their performance difficult to analyse.

Algorithms often have many tuning parameters which may critically affect system performance. Any characterisation must also take these into account.

Ground truth is expensive to acquire and is always open to interpretation. Just because a human operator indicates the position of a target does not mean it is the true position.

Testing is time-consuming and is often not recognised as valuable research, particularly in academic publications. For example, in a recent conference (ICPR '98), only six papers out of 494 directly addressed performance assessment issues. Most of the remainder presented new algorithms or techniques, and only a quarter of these provided any comparisons with existing algorithms. Less than a fifth analysed performance in a quantitative fashion beyond this.

These factors combine to make the task of IP system performance characterisation a challenging but important area of research.

1.3 Scope of this Research

This thesis goes some way towards addressing some of the deficiencies in IP system development and attempts to overcome some of the main difficulties in performance characterisation that have been outlined above. It contains a new methodology or ‘rule book’, which is intended to assist engineers and researchers in developing IP systems and algorithms. Although techniques have been developed for assessing individual algorithms, e.g. edge detection, segmentation etc., the author believes that no one has to date developed a generalised methodology for IP performance assessment for the system user. This new methodology will:

- Enable the analysis and performance predictions for a wide range of IP systems, even when they are applied to a variety of problems.
- Be in many cases considerably less time-consuming than the equivalent building and testing of the algorithm, and give results with an appropriate degree of accuracy.
• Guide developers in analysing IP problems, decoupling the system from the operating conditions, gathering the appropriate data, then categorising the problem according to a set of criteria developed here.
• Describe a new technique for modularising IP systems, and for considering the system as a series of boxes, which propagate performance or quality characteristics in addition to data.
• Show how the interaction between the different modules and the parameters can be analysed.
• Demonstrate how the analysis of each of these modules can then be considered as a relatively simple transfer function between operating conditions and the module performance.
• Guide the developer in how to combine these transfer functions to derive overall performance estimates for a system under different operating conditions.
• Provide the building blocks for universal performance data predictors that can accompany off-the-shelf algorithms.
• Supply the necessary performance data which could then be used to optimise system performance.

The thesis demonstrates the effectiveness of the new methodology with a detailed assessment of four real-life IP problems, and a demonstration of how it could be applied to a further two problems. These show how the methodology can be used to analyse a variety of IP tasks, and how in practice it achieves the goals set out above.

1.4 Structure of this Thesis

The remainder of this thesis, consisting of eight chapters, is structured as follows:

Chapter 2 introduces the reader to the principles behind performance measurement in the field of IP. It reviews the existing work in the field and surveys the different approaches taken by previous researchers. It then describes the different IP and other techniques which are used in subsequent chapters.
Chapter 3 describes the new methodology in depth. It describes the steps which an IP system developer should take when developing and evaluating a system. This forms the structure of chapters 4-8, which apply the methodology to the performance analysis of real-world industrial IP problems.

Chapter 4 contains the performance analysis of the first of the IP applications. It uses a semi-empirical approach to evaluating an existing algorithm for use in tracking ladles of molten steel in a steelworks. It gives estimates of the likely final performance of a vision system used to solve this problem, and shows how the problem may not be amenable to an IP solution.

Chapter 5 is a mainly theoretical performance analysis of a second IP system for use in a steelworks, for identifying batches of steel. It includes a new error-correcting code developed by the author for this application.

Chapter 6 analyses the performance of a system for determining the aberrations in transmission electron microscope (TEM) lenses. It describes a new theoretical model of the image and the algorithm for calculating the errors in the estimates of the aberrations. The theoretical model was developed in collaboration with the author's supervisor, the application of the methodology is the authors work alone.

Chapter 7 analyses the performance of a vision system which is currently in operation on a self-guided vehicle manoeuvring drums of nuclear waste around a storage plant.

Chapter 9 concludes the thesis with a discussion of the research and suggestions for further research.

Appendix A gives a more detailed discussion of the error correcting codes used in chapter 5 and also a description of a new code design which was developed for this project.
Appendix B demonstrates how the methodology can be applied to two other IP problems, but does not analyse them in depth. It demonstrates the general applicability of the approach to a wide variety of vision problems.
Chapter 2: Background

This chapter describes the current state of the art in measuring the performance of image processing systems, and outlines the different approaches that have been adopted. It shows where this thesis fits into current research, insofar as it represents the first attempt at a complete methodology for performance characterisation. The work described in this thesis is intended for image processing system developers who are interested in evaluating the performance of systems they design. The chapter finishes by describing some of the image processing techniques which will be used later in the thesis.

2.1 Performance Measurement in other Engineering Disciplines

Performance measurement in other engineering disciplines is highly developed. An engineer will choose the best specified product for a task, as it can often be determined without testing whether the product is a suitable candidate for the task in hand. This choice is often achieved using the following techniques:

1. Standardisation
   Using standard fittings, sizes, interfaces etc., in order to avoid compatibility problems.

2. Modularisation
   Breaking the system down into smaller functional blocks allows the performance of these blocks to be analysed individually and then combined to predict the performance of the system as a whole.

3. Theoretical Models
   The performance of many engineering systems is evaluated using theoretical expressions describing the behaviour of materials, fluids, components etc. and can be evaluated by hand or numerically by computer.

4. Testing
   Systems which are not amenable to theoretical analysis are often accompanied by data forming an empirical ‘performance envelope’ which describes the performance under a
A variety of conditions. These may be taken from tests of the actual devices operating in situ, or from scale models, simulations in wind tunnels etc.

In the field of image processing, the first technique is being tackled by IP researchers [5] and is not being addressed in this thesis. However techniques 2, 3 and 4 are often taught to engineers during training. In some disciplines, such as computer aided engineering, these procedures are formalised further, by the development of methodologies for specifying, designing and breaking down engineering problems [6-9]. There have also been similar tools developed for assessing the performance of computer code [10]. The techniques of modularisation, theoretical modelling and testing inspired much of the methodology development that is described in the remainder of the thesis.

2.2 Different Techniques for Measuring Performance

Three main techniques for measuring performance of image processing algorithms and systems have begun to be developed. These are performance characterisation, performance evaluation and benchmarking. This section describes the differences between them and how work in each field has developed so far.

2.2.1 Performance Characterisation

Performance characterisation is usually defined in IP to mean the measurement or prediction of the performance of an algorithm or system throughout the full space of the expected operating conditions. This performance characterisation, although time-consuming, can then be used to predict performance when a system is used for a new application or under different imaging conditions. Different methods for performance characterisation, such as testing and analytical algorithmic modelling have been developed, and are described in sections 2.2.1.1 to 2.2.1.3.

Performance characterisation has already been applied to many algorithms or classes of algorithms for specific IP tasks. One example involved adding noise to and obscuring parts of real images, and measuring the variation in the performance of recognition algorithms [11]. There have also been similar attempts at analytical modelling of other low-level algorithms, such as edge detectors [12] [13, 14], texture
segmentation [15, 16], image stabilisation [17], and algorithms such as pixel vectorisation [18], binarisation [19], and detection algorithms [20].

2.2.1.1 Algorithmic Modelling

A formal description of algorithmic modelling was given in [21] as a way of characterising algorithm performance. Algorithms are developed in conjunction with a mathematical (analytical) model of the algorithm that describes their performance. This model can then be used as a transfer function, to calculate performance, given the input operating conditions. For example, analytic models of line and circle fitting algorithms have been developed, which enable the estimation of errors when input variables such as line length and noise are varied. These predictions are then compared with results from empirical tests on synthetic data [22]. Other models have been used to estimate various location and detection errors in corner detection algorithms as noise levels vary [23].

One of the most important features of algorithmic modelling, which is not addressed in the literature but forms an important part of this thesis, is parameter determination. The effectiveness of the algorithm model depends critically upon selecting the appropriate parameters to describe the input conditions and performance metrics. Selection of parameters is discussed in chapter 3.

Because performance can be determined from the algorithm model as a function of operating conditions, algorithmic modelling is a very useful tool for performance characterisation. If the operating conditions change, the algorithm model is (usually) still valid. However, one of the limitations of algorithmic modelling is that it cannot be carried out if the input conditions cannot be described quantitatively. It can also be time consuming. The concept of algorithmic modelling is used in the industrial IP task analyses later in this thesis.
2.2.1.2 Performance Envelope Measurement

The performance envelope specifies the performance of a system as the parameters describing the operating conditions are varied. It is a surface in a multidimensional space, with each dimension corresponding to an input parameter or performance metric. A simple one-dimensional example is the speed/altitude capability of an aircraft. The performance envelope can either be calculated using the algorithmic modelling method described in section 2.2.1.1, or be measured by using the testing techniques discussed in section 2.2.1.3. Either method requires appropriate parameter selection to ensure the dimensions in the space correspond with the most important measures of the input conditions.

Once calculated, either theoretically or empirically, the performance envelope can then be used to determine the performance of a system under known operating conditions.

2.2.1.3 Testing

One method of evaluating performance is simply to implement the system and test it on real data. As this thesis will show, under some conditions this may be necessary. Haralick [24] proposed that both performance evaluation and characterisation can be carried out by describing the ‘normal’ operating conditions (i.e. input data), then randomly perturbing the operating position and measuring the effect on performance. However the probability distribution of the actual operating conditions is critical to overall system performance, and must be taken into account.

Testing can also be time consuming, as the algorithm must be implemented and a set of input data with the appropriate variation in operating conditions must be acquired. However, reductions in the time and expense involved in testing can be achieved via modularisation, as discussed in section 2.2.4. The use of modularisation has the advantage that only those components of the algorithm that cannot be analysed readily using other methods need to be tested in this way. Often the requirements for testing have more to do with the complexity involved in describing the operating
conditions, than with the algorithm itself. A discussion of the use of testing for performance characterisation is given in [25].

2.2.2 Performance Evaluation

Performance evaluation differs from performance characterisation in that it is only trying to measure performance against a pass/fail criterion. This means that for a given algorithm, only a subspace of possible operating conditions needs to be analysed, namely those conditions under which the algorithm will be operating under when performing the task for which it is being considered. It also means that certain performance characteristics, which it would be necessary to measure for complete characterisation, can now be neglected. The advantage of performance evaluation over characterisation is therefore ease of implementation. However it does not give a complete description of system performance and therefore is not valid if the operating conditions change or if the system is used for a different task.

2.2.3 Benchmarking

Benchmarking differs substantially from the previous two techniques. Benchmarking implies a common, ‘level playing-field’ test, whereby an identical task is carried out with a variety of algorithms and each is given a performance measure. A familiar example from outside the field of IP is the DOT fuel consumption test, where every vehicle is tested for fuel consumption under a specified set of driving conditions and given a figure of performance.

2.2.3.1 Examples of Benchmarking

Benchmarking techniques have already been applied, with varying degrees of success, to some image processing tasks. Many papers have been written on comparing algorithms for specific IP tasks [16, 26-31]. Methodologies have been developed to apply benchmarking techniques [32]. The Abingdon Cross survey was one of the earliest and most famous of any attempts at benchmarking in the field of image processing [33]. It consisted of a pre-defined image on which a standard set of image processing operations had to be performed. Although important, the Abingdon Cross’s principal
aims and achievements were to compare the parallel computers on which the algorithms were run, rather than to compare the algorithms themselves.

2.2.3.2 The “StatLog” Project

The ESPRIT-funded StatLog project, which ran from 1990-1993, was designed to provide a quantitative comparison of a wide range of classification algorithms, by testing their relative performance under ‘level playing field’ conditions against a wide range of classification tasks, although only a few of these used image databases. The classification algorithms fell into three main categories - statistics-based, machine learning based, and neural networks. The results of this project are summarised in a book on the project [34].

Statistics-based algorithms are generally considered to have an explicit underlying probability model, and it is usually assumed that the algorithms can be ‘tuned’ by humans (mainly statisticians), who can control the overall flow of the algorithm. Machine learning (ML) is based on earlier artificial intelligence ideas, which usually try to construct decision trees (‘if-then’ rules) based on the supplied training data. The term ‘neural networks’ covers a wide range of algorithms, but is usually taken to imply a network of one or more interconnected layers of nodes (‘neurons’) which are trained to adjust their contents in response to repetitive presentation of a set of training patterns. The training may be supervised or unsupervised. Supervised learning, requires the user to input the class of each training pattern. Unsupervised learning extracts information about the training data set without explicit guidance during training, but the user has subsequently to specify which system response corresponds to which category.

The StatLog project provided an excellent review of these different approaches and provides a role model for any benchmarking studies, whether in the field of image processing or in other areas where classification is needed.

Benchmarking in general suffers from the big disadvantage that it only measures system performance under a specific set of operating conditions. Thus if the conditions under which the benchmark test was carried out are not representative of the conditions
under which the system will be used, the results will be inaccurate as a predictor of system performance. However benchmarking does allow a direct comparison between algorithms. Developers often find this idea attractive as they want to know which algorithm is the ‘best’. Unfortunately performance depends critically on the operating conditions, so that without a very careful choice of tests, such comparisons can be misleading.

Therefore for the reasons described above, benchmarking will not be considered further in the remainder of this thesis.

2.2.4 Modularisation

The modularisation of algorithms has been suggested as an aid to performance characterisation [4, 35, 36]. Breaking down the algorithm allows individual sections to be analysed. These less complex sub-systems can then be analysed, and their performance characteristics combined with those of the other modules to yield a final performance measure. One of the most important aspects of modularisation is the use of quality metrics, which determine how the performance characteristics of each module are propagated through the system. This is described briefly in [35], and is developed in more depth in chapter 3.

Modularisation greatly facilitates the analysis of IP system performance. Algorithm modules are usually less complex than the system as a whole, and are consequently easier to model. Modularisation also enables different sub-system algorithms to be included in the overall system, and the performance analysis can be updated readily to measure the affects of the change. Methods for system modularisation are developed in chapter 3, and demonstrated extensively in chapters 4-8.

2.3 Related Work

Several other techniques have been developed which are related to IP system performance analysis. These include standardisation of frameworks and image databases, automatic algorithm tuning, statistical testing techniques and also the lessons learnt from actual algorithm development.
2.3.1 Standarisation

Attempts to impose standards in image processing have followed two main lines of development: standardising the frameworks for development and testing and standardising databases of test images.

2.3.2 Standard Frameworks

2.3.2.1 Image Understanding Environment (IUE)

The IUE is a five-year US program, sponsored by the Defence Advanced Research Projects Agency (DARPA), to develop a common software environment for the development of algorithms and application systems [5]. Its goals are to improve research productivity, to provide a standardised format for education and development, to standardise computational models and to improve technology transfer. It can be used as a basis for algorithm development. DARPA had earlier produced a ‘DARPA benchmark’ - a set of test images (of moving, overlapping rectangles) which was used as a test piece for comparing the relative performance of various algorithms.

2.3.2.2 Image Processing Standards/BSI Collaboration

The standards organisations ISO and IEC set up a joint technical committee to investigate the possible provision of standards for image processing. A working group (ISO/IEC JTC 1/SC 24/WG7) failed to reach a consensus on the correct approach to take.

2.3.2.3 Harness for Algorithmic Testing and Evaluation (HATE)

The HATE project is a tool developed primarily by Clarke [21] which presents an environment for the creation of a set of common tests which will be universally applicable to assessing the performance of various algorithms. HATE runs tests and accumulates data for pooling in a central repository, thus allowing a comparison of different algorithms.
2.3.2.4 Standard Image Databases

A different approach to comparing algorithm performance is to provide databases of standard images, covering a wide range of operating conditions against which algorithms can be tested. Several of these exist, such as the National Institute of Standards in Technology (NIST) database of handwritten zip codes, British Aerospace’s segmentation databases [37] and many more are being developed. Examples include databases for face recognition tasks [38], automated manufacturing [39] and more general image databases [40].

2.3.2.5 Automatic Algorithm Tuning

Automatic selection of algorithm tuning parameters and automatic system configuration is an area in which some related work has been carried out. Any optimising technique requires a ‘fitness’ measure, which in turn implies performance measurement. This development has yielded some novel ways of performance measurement. For example, Ramesh and Haralick [41] describe a methodology to optimise the selection of algorithm tuning parameters, by minimising appropriate performance measures. They demonstrate the technique on an algorithm for edge detection and linking. Similar techniques have been developed in other area of IP such as boundary detection [42], automatic target recognition [43] and the selection of edge detection algorithms [44].

2.3.2.6 Statistical Techniques for Algorithm Testing

Research has also been undertaken to develop novel statistical tests for evaluating IP systems. These include self-consistency [45], a novel method which eliminates the need for ground truth to accompany data. Other techniques, such as bootstrapping, enable smaller data sets to be resampled to simulate larger ranges of data for testing [46-49]. Research work investigating the effect of quantisation errors [50, 51] which are introduced when images are digitised, has been carried out. There are also theories concerning the behaviour of statistics in the extremes of distributions [52, 53] This is where much of the error analysis is carried out, and is one important consideration when estimating the probability of unusual events with measurement and
modelling errors [54]. Also important is the effect of different non-linear functions on the propagation of variance [55].

2.3.2.7 Real-World Algorithm Development and Testing

Some of the more useful research work in performance evaluation has stemmed from attempts by designers to characterise the performance of real-world systems as they are developed. These have the advantage of addressing the complexity often present in real-world IP problems [56-59] [43]. Some have considered the different modular stages in an algorithm [60], though have not attempted to analyse their performance in-depth.

Image Characterisation and Performance Metrics

An important feature of any characterisation or evaluation procedure is the necessity to measure the appropriate characteristics of the input data. This has been attempted for several types of algorithm and problem [61-63]. There have also been attempts to develop appropriate performance metrics for different IP tasks such as image segmentation [13, 37, 64], and also in related fields such as non-destructive testing, where the probability of detection (POD) is used to measure performance as a function of defect size [65]. One of the most useful measures developed is the receiver operating characteristic (ROC) curve. This plots the probability of false indications (PFI) against the POD as a function of some classification threshold [66]. Although this yields a curve in two-dimensions, it can be turned into a useful one-dimensional performance measure by integrating the area under the curve [67].

2.4 Background on Techniques

During the development of the methodology and analysis of the industrial examples, several different IP techniques were investigated, implemented and analysed. The following sections describe the various techniques used. The first is an outline of a simple matching technique, called template matching, which is used in several of the examples. The second involves tracking using the adaptive Kalman filter, a technique which is not used in the full analysis, but is included here as an illustration of how the ladle tracking system could have been made more sophisticated. The third section
describes some of the different techniques which have been developed in an attempt to determine the lens aberrations in transmission electron microscopes (TEMs), the problem analysed in chapter 6. There is then an introduction to the theory of error-correcting codes, which are used in chapter 5 in ladle identification. Different algorithms that could have been used in chapter 7 on drum location are not described here, as the algorithm had already been designed by the plant operators. Their system design is described in chapter 7.

2.4.1 Template Matching

Template matching based on correlation is used for determining whether and where a specific reference pattern (the template) is located within an input image (the scene). It is typically used for detecting and locating objects of known sizes and orientations in scenes.

If \( t(i,j) \) refers to the pixel grey level at position \((i,j)\) in the \( M \times N \) template image and \( s(i-m,j-n) \) refers to the corresponding grey level in the scene image when the template is displaced by some distance \((m,n)\). The difference correlation measure or score at position \((m,n)\) is then calculated according to [68]:

\[
D(m,n) = \sum_{i=m}^{m+M-1} \sum_{j=n}^{n+N-1} \left| t(i,j) - s(i-m,j-n) \right|^2
\]

\[
\text{eq(2-1)}
\]

Expanding this yields the form:

\[
D(m,n) = \sum_{i=m}^{m+M-1} \sum_{j=n}^{n+N-1} |t(i, j)|^2 + \sum_{i=m}^{m+M-1} \sum_{j=n}^{n+N-1} |s(i, j)|^2 - 2 \sum_{i=m}^{m+M-1} \sum_{j=n}^{n+N-1} t(i, j)s(i-m, j-n)
\]

\[
\text{eq(2-2)}
\]

The first term is a constant for a given template, the second varies if the mean grey level intensity varies across the scene image. The third term is a cross correlation term which also varies with mean grey level. Thus to compensate for variations in scene grey level, the normalised difference correlation measure can be calculated as:
or the normalised cross correlation coefficient:

\[ C(m,n) = \frac{\sum_{i=m}^{m+M-1} \sum_{j=n}^{n+N-1} t(i,j)s(i-m, j-n)}{\sqrt{\sum_{i} \sum_{j} |t(i,j)|^2 \sum_{i} \sum_{j} |s(i,j)|^2}} \]

eq(2-4)

The Cauchy-Schwarz inequality states that:

\[ \left| \sum_{i} \sum_{j} t(i,j)s(i-m, j-n) \right| \leq \sqrt{\sum_{i} \sum_{j} |t(i,j)|^2 \sum_{i} \sum_{j} |s(i,j)|^2} \]

eq(2-5)

Equality holds if and only if:

\[ t(i,j) = \alpha r(i-m, j-n), \quad i=m,...,m+M-1 \quad j=n,...,n+N-1 \]

eq(2-6)

where \( \alpha \) is a constant. Hence \( c(m,n) \) is always less than or equal to unity and reaches a maximum value of unity if the template is an exact, replica of the scene at \( (m,n) \). A variety of templates enables this technique to be used to search not only for the location of an object, but also to identify the size and type of object visible in the image.

### 2.4.2 Tracking

Sophisticated tracking techniques are not used in the analysis of the industrial problems in the following sections, as simple techniques, although less effective, are easier to model and their detrimental effect on performance under the chosen operating conditions is expected to be minimal. However one classic tracking technique is described here briefly and its possible use in this work discussed.
The linear Kalman filter is a classic tool of optimal estimation theory [69]. It is based on a model of a physical system involving a time dependent state vector, $x$, and a set of linear equations called the system model. The state vector contains enough variables to describe the dynamic properties of a system. In the case of tracking an object in three dimensions, velocity and displacement in each direction are sufficient. The system model describes the change in state over time. If we sample at equally spaced time periods, $t_k = t_0 + k\Delta T$, with $k = 0, 1, \ldots$ and $\Delta T$ a sampling interval. The linear system can be modelled in vector form

$$x_k = \Phi_{k-1}x_{k-1} + \xi_{k-1}$$

where $\xi_{k-1}$ is a vector indicating random additive noise. The state transition vector $\Phi_{k-1}$ is also a function of time to allow for complex system dynamics. A noisy measurement is made of the state vector at time $t_k$ and the Kalman filter theory assumes that the following relation holds:

$$z_k = H_kx_k + \mu_k$$

where $z_k$ is the vector of measurements taken, $H_k$ is the measurement matrix and $\mu_k$ a random vector modelling additive noise.

If $\mu_k$ and $\xi_k$ are white, zero-mean, Gaussian processes with covariance matrices $Q_k$ and $R_k$ the Kalman filter can be shown to be the optimal estimate of the state of the system [69].

The Kalman filter is implemented by evaluating the following recursive equations:

$$P'_k = \Phi_{k-1} P_{k-1}\Phi_{k-1}^T + Q_{k-1}$$

$$K_k = P'_k H_k^T (H_k P'_k H_k^T + R_k)^{-1}$$
\[ x'_k = \Phi_{k-1}x'_{k-1} + K_k(z_k - H_k\Phi_{k-1}x'_{k-1}) \]

\[ P_k = (I - K_k)P'_k(I - K_k)^T + K_k R_k K_k^T \]

eq(2-9,10,11,12)

where \( x'_k \) is the optimal state estimate at time \( t_k \). \( P_k \) is the \textit{a priori} estimate of the error covariance and \( K_k \) is the gain or blending factor that minimises the \textit{a posteriori} error covariance.

If \( x \) is a 2-D state vector \([x_1, x_2]^T\), the region of the plane centred around \( x'_k \) which contains the true state with a given probability \( c^2 \) is the ellipse

\[ (x - x'_k)(P_k)^{-1}(x - x'_k)^T \leq c^2 \]

eq(2-13)

The axes of this ellipse are \( \pm c\sqrt{\lambda_i e_i}, \; i = 1, 2 \), where \( \lambda_i \) and \( e_i \) are the eigenvalues and eigenvectors, respectively, of \( P_k \). This uncertainty ellipse can then be used to calculate the search space for the next frame in the tracking sequence.

For the example tracking problem discussed in chapter 4 however, the time dependency of \( \Phi_k \) is not known, due to the sudden changes in acceleration of the object. \( \Phi_k \) could model the inertia of the object being tracked, i.e. assuming no acceleration, which would provide a good estimate of state during relative simple motion. However changes in acceleration are frequent enough in this application that the improvement in system performance is likely to be secondary to other factors. For this reason, and to simplify some of the analysis, the use of a Kalman filter is not considered in chapter 4.

2.4.3 Transmission Electron Microscopy Lens Aberration Analysis Techniques

Aberrations in the lenses of transmission electron microscopes (TEMs) limit the resolution that can be achieved [70] and the accuracy of measurements taken using them [71]. Several different image processing techniques have been developed to analyse the diffractograms generated by TEMs, which contain information that can be used to determine, and correct for, some of these aberrations. The full IP problem is described in chapter 6, where the use of template matching and the Orientation Correlation Function (OCF) are analysed in depth. However a brief summary of the IP task is presented here,
and some of the other techniques for lens aberration determination which have been
described in the literature are discussed.

The TEM can be used to generate a diffractogram, see Figure 2-1, the intensity
of which describes a function of the form (from [73]).

\[
F(r, \theta) = \sin^2(\pi \lambda^3 C_3 r^4 / 2 - \pi \lambda C_1 r^2 / 2 - (\pi \lambda A_1 r^2 / 2) \cos(2(\theta - \phi_{22})))
\]

eq(2-14)
The reasons for the function having this form are explained in chapter 6.

The image processing problem is to fit this function to the noisy data in the
diffractogram and hence deduce the values of the parameters for the spherical
aberration, \( C_3 \); defocus, \( C_1 \); two-fold astigmatism, \( A_1 \); and the angle of primary
astigmatism, \( \phi_{22} \). (The actual aberration determination procedure is more complex, as
other aberrations exist which must be measured by producing images with an induced
tilt in the electron beam. This is discussed in chapter 6.)

2.4.3.1 Manual Fitting

One approach is simply to generate a surface using estimates of the parameters,
compare it side by side with the real diffractogram, and then adjust the parameters
manually until the images appear to match [72] as shown in Figure 2-1.
Figure 2-1 A diffractogram from a transmission electron microscope. The top-right of the image shows the original diffractogram, the bottom left a synthetic diffractogram which has been manually fitted to the data.

2.4.3.2 Stretch Remapping

One possible automated solution is to integrate the signal strength tangentially around the diffractogram. This improves the signal to noise ratio and can be achieved quite easily when the diffractogram is circular, i.e. when the astigmatism is small. However when astigmatism is significant, other techniques must be employed. One such method is sector averaging, as proposed in [73]. This involves assuming that the diffractogram is approximately circularly symmetrical over a small angle and integrating tangentially through this angle (an angle of 36° was chosen for this particular diffractogram as an optimal compromise between maximising the number of data points used for the integral and minimising the variation in the diffractogram over the sector. Where fewer rings are visible in the diffractogram, the size of the angle has to be increased to e.g. 60°, to get acceptable errors [73]). The result can be used to find an approximation to the position of the zeroes in the function. This information can then be fed back to ‘stretch’ the diffractogram to approximately circularly symmetrical, as shown below.
From the original image a, an approximation of the radius $r$ over sector $\phi$ is made. The radius is then scaled to convert the image, assumed to approximate image b, into a circularly symmetrical diffractogram as shown in c. Although this overcomes some of the problems due to noise, it seems rather crude and error prone to approximate some of the sectors as circular.

2.4.3.3 Remapping and Fourier Analysis

A fast technique for aberration determination from diffractograms is described in [74]. This involves taking radial sections through the diffractogram, remapping it into $(\phi, r^2)$ space, and then extracting the spectrum. The technique relies on prior knowledge...
of the spherical aberration, $C_3$, to within a reasonable degree of accuracy. The principal behind the approach is that

$$\sin^2(\gamma) - \frac{1}{2} = -\frac{1}{4} \exp(i\pi\lambda C_1 r^2) \exp\left(\frac{1}{2}i\pi\lambda^3 C_3 r^4\right) - \frac{1}{4} \exp(-i\pi\lambda C_1 r^2) \exp\left(-\frac{1}{2}i\pi\lambda^3 C_3 r^4\right)$$

\[ \text{eq(2-15)} \]

Multiplication by

$$\exp\left(-\frac{1}{2}i\pi\lambda^3 C_3 r^4\right)$$

\[ \text{eq(2-16)} \]

and substitution of $K = r^2$ gives

$$\sin^2(\gamma) = \frac{1}{2} - \frac{1}{4} \exp(i\pi\lambda C_1 K) - \frac{1}{4} \exp(-i\pi\lambda C_1 K) \exp\left(-i\pi\lambda^3 C_3 K\right)$$

\[ \text{eq(2-17)} \]

Taking the Fourier Transform of this function with respect to $K$, the spatial frequency squared, should yield peaks at $K=\pm1/2\pi\lambda C_1$. Thus $C_1$ can be determined if the position of the peaks can be found and measured.

This works well on synthetic data, as shown below. It requires good estimates of the background, as this should be subtracted from the image to minimise the effect of low frequency components obscuring the true signal peak.

Figure 2-3 shows a noise-free synthetic diffractogram with defocus, $C_1$, the only aberration. Figure 2-4 is the spectrum with $K$ on the vertical axis and azimuthal angle, $\phi$, along the horizontal axis. Spherical aberration, $C_3$, is zero, so the $\exp(-1/2i\pi\lambda^3 C_3 r^4)$ term is unity. As expected, the image gives two sharp lines at $K=\pm1/2\pi\lambda C_1$. There is no $\phi$ dependency due to the circular symmetry of the diffractogram. Figure 2-5 is a cross-section through Figure 2-4, clearly showing the two peaks.

Figure 2-6, Figure 2-7 and Figure 2-8 show the same analysis with the inclusion of astigmatism, $A_1$. Again the lines are sharp, but they now vary sinusoidally with $\phi$. The phase shift in $\phi$ indicates the orientation of the astigmatism.
Figure 2-9 to Figure 2-12 demonstrate the effects of the inclusion of spherical aberration. Figure 2-9 is a diffractogram generated with a spherical aberration term. Figure 2-10 is the spectrum without multiplying by the \( \exp(-1/2i\pi\lambda^3 C_3 r^4) \) term. It can be seen that the \( K^2 \) term due to the spherical aberration has blurred the peaks. Figure 2-11 shows the spectrum after multiplication by \( \exp(-1/2i\pi\lambda^3 C_3 r^4) \). One of the peaks (the lower sinusoid) is sharpened; the other is blurred completely.

Figure 2-13 shows a diffractogram handfitted to real data. Its spectrum, Figure 2-14, and cross-section, Figure 2-15, show the synthetic data with the lower sinusoid clearly visible. The real data shows no such peak due to the very low signal to noise ratio when one radial slice at a time is taken. Thus for this example, the technique requires either a high signal to noise ratio, a low astigmatism so that radial averaging is possible, or some knowledge of the astigmatism that might permit radial averaging.
Figure 2-3 Synthetic diffractogram with defocus only

K

azimuthal angle, $\phi$  $\pi$

Figure 2-4 Frequency spectrum of the diffractogram in figure 2-3

Figure 2-5 Cross-section through the spectrum in figure 2-4.

K

2.80E+07

1.40E+07

0.00E+00

$K$

Figure 2-6 Diffractogram with defocus and astigmatism (same scale as figure 2-3)

K

azimuthal angle, $\phi$  $\pi$

Figure 2-7 Frequency spectrum of the diffractogram in figure 2-6 (same scale as figure 2-4)

Figure 2-8 Cross-section through the spectrum in figure 2-7

3.60E+07

1.80E+07

0.00E+00

$K$
Figure 2-9 Synthetic diffractogram with defocus, astigmatism and spherical aberration (same scale as figure 2-3)

Figure 2-10 Spectrum of figure 2-9 before multiplication by \( \exp(-1/2i\pi\lambda^3C\gamma^4) \) (same scale as figure 2-4)

Figure 2-11 Spectrum of figure 2-9 after multiplication by \( \exp(-1/2i\pi\lambda^3C\gamma^4) \) (same scale as figure 2-4)

Figure 2-12 Cross-section through spectrums in figures 2-11 and 2-12

Figure 2-13 Real diffractogram (top right) with hand-fitted synthetic diffractogram (bottom left)

Figure 2-14 Spectrum of figure 2-13 after multiplication by \( \exp(-1/2i\pi\lambda^3C\gamma^4) \) (same scale as figure 2-4)
Figure 2-15 Cross-section through spectrums in figures 2-14
2.4.3.4 Template Matching

The template matching technique described in the previous section can also be used to determine the best fit [75]. Synthetic diffractograms are generated with different values of the parameters $A_1$, $C_1$, and $\phi_{22}$, and the sampling frequency and $C_3$ held constant. These are then correlated with the real diffractograms using the normalised correlation measure described in section 2.4.1. As the location of the diffractogram in the image is already known, the operation only needs to be performed once for each synthetic image. The values of the parameters in the input image are then assumed to be same as those in the synthetic diffractogram which gave the best correlation match.

2.4.3.5 Finding Axes of Symmetry

With any of the techniques described above the search space can be reduced by first finding the orientation of the diffractogram. A particularly robust way of doing this is using the method described in [76]. The diffractogram is separated into two halves along an arbitrary axis. One half is then mirrored and correlated with the other. This is then repeated along different axes and the correlation measures, known as the Orientation Correlation Function (OCF), are logged. The best match correlation scores should correspond to the axes of symmetry of the shape; that is the major and minor axes quasi-elliptical diffractogram. This then allows $\phi_{22}$ to be determined, with only an uncertainty in differentiating between the major and minor axes, or $\phi_{22}$ and $\phi_{22}+\pi/2$. 
Figure 2-16 Illustration of the technique for finding the angle of two-fold astigmatism for a TEM diffractogram described in [75]. The diffractogram is inverted along a series of different axes, and then correlated with itself. There should be resulting diffractogram maxima (or minima) corresponding to the axes of symmetry in the image.

There have also been methods developed to measure the lens aberrations using techniques other than diffractogram analysis. These include the use of Beam-Tilt Cross Correlation [77]. These techniques are not considered in detail here.

2.4.4 Error Correcting Codes

Error-correcting codes are used in the ladle identification algorithm developed in chapter 4. They are introduced briefly here and described in more depth in appendix A.

The concept of error correcting codes was first introduced by Hamming [78] to improve the reliability of computers of the time. It relies on the code (usually binary) having a degree of redundancy built in which can be manipulated to maximise the probability of the original codeword being recovered if the received data is corrupted. The receiver checks if the received word is a valid codeword, and if not replaces it with the most likely codeword before corruption. It is important at this point to introduce a few concepts and some notation [78, 79].
2.4.4.1 Hamming Distance

The Hamming weight of a word is the number of nonzero digits in the word. So if words of length \( n = 5 \) are being used, the weight of 01101 would be 3. This is denoted by \( \text{wt}(01101) = 3 \). The Hamming distance between two words is the number of places where the digits differ between the two words. For example, 01101 and 01100 differ only in the last digit so the Hamming distance, or distance, between the first and second is 1. This is denoted by the letter \( d \). Here \( d(01101, 01100) = 1 \). To correct errors we will check to see that we have received a codeword. If the received word is not in the code, we can replace it with the nearest codeword using the distance function just defined. The following diagrams illustrate the concept of distance. Both of the illustrations below are from [80].

![Diagram of Hamming Distance](image)

**Figure 2-17 Illustration of the concept of distance in code of length \( n = 3 \). The codewords are the spheres on the vertices, here with distance \( d = 2 \), as they would have to be displaced along two axes before they would occupy the same vertex as another codeword.**

In the Figure 2-17 codewords of length \( n = 3 \), are placed on the vertices of the unit cube. The black spheres represent the codewords. In this code there are the four code words 000, 101, 011, and 110, corresponding to two bits of information \( (k = 2) \). If the code 111 was received, we would know that an error had occurred but could not correct for it as it is equally likely that the original word was 101, 110, or 011 (as each is the
same distance from the received codeword). This is an error detecting code but could not be an error correcting code.

![Illustration of distance in code of length n=3](image)

**Figure 2-18** Illustration of the concept of distance in code of length $n=3$. Here, the codewords have distance $d=3$, as this time they would have to be displaced along three axes before they would occupy the same vertex as another codeword.

In this diagram only 000 and 111 are code words, so $k=1$. If 000 was transmitted but one error occurred in the channel either 100, 001 or 010 would be received. In this case the error could be corrected for as it is more likely that one bit was in error. Therefore changing the code from 000 to 100 is more likely to recover the original signal than changing it from 111 to 100. The important difference between these two diagrams is that the first has a distance of two between each pair of codewords and the second diagram has a distance of 3. In general it makes sense that $t$ errors can be detected if the minimum distance between any two codewords in a code is $t + 1$. And $t$ errors can be corrected if the minimum distance $d$, of the code is $2t + 1$. The task in error correcting coding is thus to generate codewords with a maximum coderate, that is the ratio of information carrying bits to total bits, and the maximum Hamming distance.

**Visual Versions of Error-Correcting Codes**

Error-correcting codes were originally developed for correcting bit-errors in electronic data transmission. However the principles behind them can also be applied to visual identification codes. By encoding the binary data in the code as a series of black
and white regions in an image, with black regions corresponding to ‘0’ and white to ‘1’, error-correction data can be encoded into an image. An example of this is a barcode, which encodes an error-detection code into a one-dimensional pattern. The error detection properties can be used to determine if the barcode was scanned correctly, and if necessary, indicate that the barcode should be scanned again. By using a two-dimensional pattern, imaged with a video camera, more information and more sophisticated error correction can be included. Various forms of two-dimensional error-correcting codes are available commercially. A few of the main ones are shown below [81, 82].

![Aztec Code](image1)

![Code 49](image2)

![Code One](image3)

![DataMatrix](image4)

![MaxiCode](image5)

Figure 2-19 Some of the commercially available visual identification codes with error-correcting capabilities.
These codes are generally based on Reed-Solomon (R-S) codes. R-S codes are a particular type of Bose-Chaudhuri-Hocquengen (BCH) code, which are a subset of the set of cyclic codes with a generator polynomial $g(x)$ chosen such that the maximum code rate is achieved [78].

2.4.5 Types of Error-Correcting Codes

Two main techniques have been developed for generating codes based on these principals, linear codes and cyclic codes. Linear codes are generated using a set of $k$ linearly independent vectors, $v_1, \ldots, v_k$ which from a generator matrix, $G$. Multiplying the message vector, $m$, by the generator $G$ yields a set of codewords $x$, where

$$x = m_1 v_1 + \ldots + m_k v_k$$

eq(2-18)

By appropriate choice of generator matrix, $G$, it is possible to maximise the code rate and error-correction rate.

Cyclic codes are generated slightly differently. An $n$-bit code is cyclic if $x = [x_n x_{n-1} \ldots x_1]$ is a codeword whenever $[x_0 x_1 \ldots x_{n-1} x_n]$ is also a codeword. Usually a polynomial form is used for the generation of cyclic codes. A generator polynomial $g(x)$ is chosen according to [78]:

$g(x)$ is the generator polynomial for a linear cyclic code of length $n$ if and only if $g(x)$ divides $1 + x^n$.

A codeword $c(x)$ corresponding to a message, $m(x)$ is then generated using $g(x)$ according to:

$$c(x) = m(x)g(x)$$

eq(2-19)

Cyclic codes can be generated which are robust to specific types of errors, such as burst-errors [83], where the errors are expected to occur in blocks or clusters, and array codes robust to column erasures [84]. They can also be designed to be robust to synchronisation errors, where the start and finish position of the code is not known [85-89]. Appendix A contains a detailed description of the operation of both linear and cyclic codes, and a description of a new cyclic code developed for the ladle identification task described in chapter 5.
2.5 Summary

This chapter has described the different research work that has been carried out on analysing the performance of image processing systems. The main techniques that have been developed, namely performance characterisation, performance evaluation and benchmarking were described and their advantages and disadvantages compared. Other related work was also discussed, such as standardisation, testing and algorithm tuning. The chapter then gave an introduction to some of the IP techniques that have been developed that are related to the performance analysis carried out in subsequent chapters. Some of the techniques are used directly in the performance analysis. Others are algorithms for solving similar problems to those in the performance analysis examples. These are provided for information and are not developed further in the remainder of the thesis.
**Chapter 3: Methodology**

This methodology is a novel way to formalise the development of image processing systems. The aim to use it as a rule book for developing, analysing and evaluating image processing systems in a way that is robust, repeatable and provides performance estimates. Two of the techniques which are used, namely covariance propagation [36] and modularisation [4], have been described previously. However neither technique has been developed to the extent to which a system developer could easily use them.

### 3.1 Typical industrial system development

In order to show where this methodology fits into the development cycle of a practical image processing system, it is necessary to describe the interaction of the different parties involved. Typically three or more parties are involved in the development of the system at three different levels. These parties are the user or customer, the system development engineer or implementer and the algorithm developer.

- The user generally has a problem for which he/she requires an IP based solution. The user will have certain requirements or specifications which the IP system is expected to meet. It is assumed that the user is familiar with the problem and with the conditions under which the system will operate. The user is not assumed to be knowledgeable about IP.
- The system development engineer starts with the user’s specification for the system, and develops an implementation, usually based on IP algorithms which are available either in the literature or in commercial IP packages. The system engineer uses his/her knowledge of IP to configure the system from a toolbox of algorithms.
- The algorithm developer is usually involved in researching new IP techniques, with a view either to publication or to inclusion in IP packages. These algorithms are assumed to be developed for general applications and are not problem specific.
We make the distinction here between an *algorithm*, such as a segmentation algorithm, and a *system*, which is a combination of algorithms, designed to perform a particular task under specific conditions.

### 3.2 Performance analysis methodology

This methodology is mainly intended to be used by the system development engineer. It breaks the process down to three different levels. The *problem* level analysis refers to the user’s specification and to the external parameters affecting the system. It guides the system engineer in gathering the appropriate data for the performance analysis. The *system* level refers to the configuration of the various algorithm components to produce a solution to the problem. The *algorithm* level refers to the sub-components or modules which are configured to produce the system.

A typical imaging system may be represented at the top level by the stages shown in Figure 3-1.

![Figure 3-1 Block diagram showing the steps in the formation and analysis of an image [37]](image-url)
This methodology is concerned primarily with the final Vision Process step. Although the factors affecting the imaging process, such as the spatial and level quantisation and camera PSF are important, they are only analysed implicitly, as we assume that the image or images are already available.

The intention is that the methodology provides a set of rules or framework such that the system designer:

- collects the appropriate information from the user
- develops the system in a structured way
- can evaluate the performance of the system as it is being developed, in order to ensure it can meet the user’s specification
- can measure the effect of tuning parameters
- ultimately uses the optimum choice of algorithms

Our description of the methodology is structured as follows:

**Introduction**
1. Description of the effect of operating conditions on the performance
2. Decoupling of operating conditions and performance
3. Parallel performance analysis sequence

**Problem Level Analysis**
4. Estimation of parameters describing operating conditions for the problem
5. Classification of problem according to complexity
6. User performance metrics

**System Level Analysis Framework**
7. Modularization of system
8. Interaction of parameters
9. Propagation of quality measures
10. Backward propagation of performance measures
11. Simplification of analysis

**Module Level Analysis**
12. Analysis of individual modules
13. Analytical analysis
3.3 Introduction

3.3.1 The Effect of Operating Conditions on Performance

The performance of any image processing system is dependent upon the conditions under which it operates. In general, it is desirable to be able to predict the performance of an algorithm under any operating conditions; a technique usually referred to as performance characterisation. The separation of algorithm and operating conditions is necessary to avoid having the situation where an algorithm’s performance is known only under certain test conditions, such as in benchmark tests. These have the disadvantage that it is not possible to predict how the algorithm’s performance will vary when the operating conditions differ from the test conditions.

3.3.2 Decoupling of Operating Conditions and Performance

The decoupling process can be illustrated by considering the performance of an algorithm as a Bayesian probability. If the algorithm performance can be described as a vector of performance metrics, $\mathbf{A}$, and if the operating conditions can be described as a vector of independent parameters $\mathbf{C}$, then the probable performance given conditions $\mathbf{C}$ is $p(\mathbf{A}|\mathbf{C})$. Hence, using Bayes’ theorem, the overall performance, $P(\mathbf{A})$ is:

$$P(\mathbf{A}) = M\mathbf{C}$$  

$eq(3-1)$

where

$$M = \begin{bmatrix}
P(A_1|C_1) & P(A_1|C_2) & P(A_1|C_3) & \ldots \\
P(A_2|C_1) & P(A_2|C_2) & P(A_2|C_3) & \ldots \\
P(A_3|C_1) & P(A_3|C_2) & P(A_3|C_3) & \ldots \\
\vdots & \vdots & \vdots & \ddots
\end{bmatrix}$$  

$eq(3-2)$

is a matrix describing algorithm performance given the operating conditions and
is the probability distribution of the operating conditions. Equations 3-1 to 3-3 assume that error probabilities due to different conditions are linear. This is a reasonable approximation when the overall error probability is low or when one operating condition is a dominant cause of errors. However when two or more operating conditions simultaneously cause high error rates, the accuracy of the estimate of the overall error rate is reduced, as the cross terms which this analysis does not consider become more significant.

In other words, the overall performance can be measured as the expected performance for a given set of operating conditions, multiplied by the probability of those conditions occurring, summed over all possible conditions. Thus when the operating conditions change, the performance can still be evaluated using the algorithm performance matrix $M$, and only the vector of conditions, $C$, needs to be re-evaluated. In many cases, of course, the factors are not independent, but they can often be treated as such in a first analysis. In most vision system development, the vision process is considered only as a transfer function between the data variables, i.e. between the raw image data and the extracted information. In this analysis, the transfer function between the parameters and the performance measure must also be evaluated. This is effectively the evaluation of the matrix $M$. To calculate the final user’s performance measure thus requires measuring or estimating $p(C)$, the probability distribution of the operating conditions.

The form of the vectors $C$ and $A$ and the matrix $M$, depends on both the problem, and on whether the data available to describe $C$ and $M$ are empirical distributions or analytic functions. In the example on ladle tracking described in detail in chapter 4, $C$ is an empirical probability distribution of only one of the image parameters, the ladle/background contrast, $\alpha$. In this case, $C$ takes the form:

$$ C = \begin{bmatrix} p(0 \leq \alpha < 5) & p(5 \leq \alpha < 10) & \ldots & p(250 \leq \alpha < 255) \end{bmatrix} $$
and $M$ has the elements

$$M = \begin{bmatrix} p(A \mid 0 \leq \alpha < 5) & p(A \mid 5 \leq \alpha < 10) & \ldots & p(A \mid 250 \leq \alpha < 255) \end{bmatrix}$$

where $A$ indicates the error rate. This yields a solution for $A$ from $A = MC$.

However where several parameters are being analysed, $C$ may have more than one dimension, and $M$ may have more than two.

### 3.3.3 Parallel Performance Analysis

The key is thus to analyse the transfer function $M$ between performance $A$ and conditions $C$ in parallel with the data analysis. This process is illustrated below.

![Parallel Performance Analysis Diagram](image)

**Figure 3-2** The parallel performance analysis of an algorithm. The algorithm can be considered as two separate transfer functions. One relates the incoming data to the outgoing information. The other is the relationship between the operating conditions and performance.

### 3.4 Problem level analysis

The first step is to estimate how complex the problem is, as this determines the appropriate level of analysis [90]. This should be carried out by the system developer in conjunction with the user, who is familiar with the conditions under which the system will operate.

#### 3.4.1 Parameter Estimation

Various external parameters which describe the formation and properties of the image may interact with the system at different stages. By external parameters, we mean
measures of the different conditions that define the appearance of the image. These are
distinct from internal tuning parameters, such as threshold level, which may be adjusted
by the user, and also from the internal quality propagation parameters which are
introduced in section 3.5.3. The external parameters might typically include lighting
levels, noise, target size etc. The first stage is to determine what these parameters are,
and estimate which of them are going to affect the performance of the Vision Process in
extracting the required information.

The system designer and/or user should:

- Specify clearly what the system is intended to do. From this one can determine an
  ultimate performance measure - see discussion below.
- Choose an algorithm, or set of algorithms, which could be used for the final
  implementation. It does not need to be an optimal design at this stage, but having an
  idea of what algorithm might be implemented enables one to decide what
  information in the scene is relevant to performance. For example, if imaging
  conditions are quite poor, the differences in performance between different
  algorithms processing the same information may well be small, compared to the
  effect on performance of the imaging conditions themselves.
- Make a list of all possible parameters which might affect the performance, bearing in
  mind that the user of the system will usually be far more familiar with the operating
  conditions than the designer. Some of these parameters will describe objects or
  features of interest, while others will relate to noise, clutter and other undesirable
  aspects of the images.
- Rank the parameters in terms of their estimated severity in producing poor algorithm
  performance. This ranking can often be carried out by judging how each parameter
  effects the ability of the human eye to perform the vision system’s task.
- Estimate or get from the user an idea of the range of each parameter. Estimate for
  each, the probability of its having particular values.
- Hence rank the parameters in terms of their probability of having values which will
  be detrimental to algorithm performance.
3.4.2 Problem Classification

There should now be enough information to estimate which parameters are not going to be significant. Poor performance of the algorithm is usually going to be dominated by the parameters which occur frequently and/or have a strongly deleterious effect. Those with relatively low probabilities of occurrence and small expected effects on performance can now be disregarded. The remaining parameters should now be classified in terms of how readily quantifiable they are. This now enables the problem to be categorised in terms of its complexity and we can make a decision on the type of performance analysis to be carried out.
3.4.3 Effect of Parameters on Image

Those external parameters that have not been disregarded can now be entered into a table that ranks them according to the severity and occurrence probability criteria. For each parameter, a qualitative estimate should be made of its effect on the input image. This should then give an indication of what quantitative measurements could be used as possible proxy measurements for the parameter. Although ambient lighting could be measured in situ using a light meter, subtle changes in illumination, position and direction could have significant effects on the captured image. For this reason, and because of the convenience of a CCD or existing video footage as a measuring device if an IP scheme is already being developed, it is often an easier and less error prone strategy to use a proxy measure which can be determined from video images. For example, the most obvious effects of ambient lighting on an image are often the contrast between the target and background and the illumination profile across the target. Both of these can be estimated quantitatively from video footage from the operating site; the contrast is simply the mean grey level difference between the target and background, and the illumination profile can perhaps be modelled as a quadratic surface fitted to the target grey level. Such a neat measurement is not possible with all parameters. Those with more complex effects on the image may need to be tested empirically. Depending on how readily the parameters can be reduced to quantitative measurements, the problem can be assigned to one of five categories:

- **Category 1.** If all parameters can be quantified accurately, the problem is in category one. A typical category one problem might be the use of synthetic images.
- **Category 2.** If all the parameters can be quantified and estimated reasonably accurately, e.g. parameters such as edge length, noise standard deviation, then it is category two. An analytical solution may be possible. A category two problem would typically be operating under well constrained imaging conditions. An example of a category 2 problem is analysed in chapter 6, involving well constrained imaging conditions in an electron microscope.
- **Category 3.** If, despite the imaging parameters being quantifiable, there are other significant non-imaging factors which affect performance, then the problem is category three. This may occur while imaging in hostile conditions where the camera
or hardware are prone to damage. An example of a category three problem is given in chapter 7.

- Category 4. If the parameters are not all quantifiable, but the imaging conditions can be reduced to a more tractable set so that a simplified model can be developed, the problem is considered category four. An example of a category four problem is the ladle tracking system operating in a steelworks, which is analysed in chapter 4.

- Category 5. If there are too many unquantifiable parameters for any reasonable approximation to be made then the problem is category 5. This category would include particularly complex problems, for example face recognition in crowds. Here the number of unquantifiable factors affecting the performance of the system, e.g. hair, make-up, lighting, shading, motion etc. make a description of the input conditions extremely difficult.

3.4.4 Performance Metrics

At the problem level, the system engineer should also determine the customer’s performance indices. These will vary from application to application, but would typically be measured in terms of number of missed defects for an industrial inspection system, mis-read characters for an optical character recognition system, etc. This will determine the elements of the performance vector, \( \mathbf{A} \).

3.5 System Level Analysis

The external parameters which have been chosen for analysis, and any internal tuning parameters, are now combined to make up the components of the vector \( \mathbf{C} \). These are all the (significant) parameters which affect the performance of the system. It is now necessary to analyse that system in order to determine the transfer function between the input parameters and output metrics (the matrix \( \mathbf{M} \)). In all but the simplest vision systems, this is not a simple task. It can be achieved by writing the algorithm, acquiring or generating sample images which span the full set of expected input conditions, and testing the algorithm on these images [24]. However this is time consuming and can be avoided for all but the most complex vision systems by system modularisation.

System modularisation can be an extremely useful and timesaving technique. The performance transfer function, \( \mathbf{M} \), for the whole system can be evaluated by
breaking the data analysis process into its constituent modules. For each of these modules, appropriate internal performance (or quality propagation) measures can be chosen. The transfer functions for each module, and the quality propagation measures between the modules, can be determined. Thus each module, \( N \), has a vector of input parameters (both external and internal) \( \mathbf{C}_N \). It also has output parameters \( \mathbf{A}_N \), related to the input parameters by a matrix \( \mathbf{M}_N \), as \( \mathbf{A}_N = \mathbf{C}_N \mathbf{M}_N \). Finally, the performance of the entire system can be predicted by propagating these performance measures through the whole system to determine the values of the final performance metrics, \( \mathbf{A} \).

### 3.5.1 Modularisation of Vision System

An initial idea for the design of the system was formulated in the previous section. The different processing stages should now be formalised and their interaction considered. In a typical system, there may be a pre-processing stage, an intermediate stage and finally a high level interpretation stage. Each of these stages or modules will usually be algorithms in their own right, e.g. edge detection, stereo matching, motion estimation. The process must now be divided into these individual processing modules for performance analysis. Each module will have its own performance characteristics which can be separated from the overall system performance. It will also propagate a data structure of some kind either to the output or to another module. A block diagram of the system can now be produced showing the ‘flow’ of data between the modules, from input image plus any other control inputs, to output data. An example is shown below from the ladle tracking problem, which is analysed in detail in chapter 4. The task is to track a ladle of steel around a steelworks. A simple template matching algorithm was chosen to test the methodology.

![Figure 3-3 Block diagram of a tracking algorithm for use in tracking steel ladles.](image)
Decomposing the system into a sequence of modules may appear to increase the initial complexity of the performance analysis. However, without breaking the system down, or resorting to the ‘build and test’ philosophy which we are trying to avoid, many complex systems are not analysable as a single unit. This approach also enables different algorithms or modules to be ‘plugged in’, e.g. different edge filtering algorithms, at each stage in the system without affecting the performance measurements taken for the other parts of the system.

3.5.2 Interaction of External Parameters

Not all the external parameters affect the algorithm at the first processing stage. For example temporal factors such as target speed will probably not affect the performance of a region linking algorithm if the algorithm is basing all its processing on only one frame. The next stage is therefore to determine which of the external parameters directly affect the performance of which modules (rather than indirectly by affecting the performance of a preceding module). These can then be added to the system block diagram, along with the final performance metric, as illustrated in Figure 3-4, from the ladle tracking problem in chapter 4. The parameters for this problem, such as contrast, obscuration, target size etc. are shown as the labels to the lower half of the block diagram.

Figure 3-4 Interaction of the parameters with the tracking system. The data being passed between the modules is shown in the upper half of the diagram. The stage at which each of the different external parameters affect the performance of different modules is shown in the lower half.
3.5.3 Propagation of Quality Parameters and Covariance

The performance of each module, except for the first, is dependent upon the performance of the preceding modules, as this will affect the quality of the input data it receives. Thus each module’s performance propagates through the system and finally affects the output data. The way these performance functions propagate through can be measured using appropriate internal quality parameters, measuring the ‘quality’ of the data which each module is passing. This is similar in concept to ‘propagation of covariance’ proposed in [36]. However a module’s performance cannot always be treated as a linear relationship between input and output errors. The next step is to determine appropriate quality measures for the output data from each processing stage or module.

3.5.4 Backward Propagation of Performance Measures

The choice of the internal quality parameters can be simplified by starting with the final performance metrics at the last module. By analysing the final module in the system, it should be possible to determine what properties of the data coming into this module will affect the final performance metric, and how these can be measured. This can then be repeated on the penultimate module, deciding what properties of the data from the preceding module affect the quality measures required by the following module. This can be repeated, propagating the error or performance measures back through the system, stage by stage, until the input parameters have been related to the output performance metric. There should now be a complete set of quality parameters which measure all the relevant properties of the data as it is passed through the system, relating back to the external parameters. These can be added to the block diagram. The quality measures for the ladle tracking example, which are the edge strength of the ladle image, the background clutter spectrum and the correlation signal depth from template matching, are shown below.
Figure 3-5 The final performance block diagram. The internal propagation parameters, namely the edge strength of the ladle image, the background clutter spectrum and the correlation signal depth from template matching, have been added. These are the full set of factors which will be used for the performance analysis.

By analysing the system backwards, i.e. considering each module in the opposite order to that in which it passes data, it is far easier to avoid analysing irrelevant variables. For example if a system only needs to determine if a particular object is in an image, a suitable performance metric might be false alarm/mis-detection rates. It would thus be unnecessary to analyse parameters which will only affect the system’s ability to determine the position of the object. However, without working backwards, it may not be possible to decide what factors are relevant in the early processing stages. Although not all vision systems are entirely sequential (they may have parallel modules and feedback loops), they are often approximately sequential, which should still enable this concept to be effective.

3.5.5 Simplification of Analysis

There should now be a reasonably complete, modularised performance analysis scheme for the system, with each module having data input and output streams, quality measure input and output streams, and external parameter input streams. However, before trying to analyse the individual modules of the system, which is the time consuming stage, it is worth at this stage trying a second iteration to improve the system analysis. Not all the external parameters which were identified at the problem analysis
stage may appear to be linked to appropriate quality measures in the system. This could be because they have been overlooked in the system analysis, or because they now appear irrelevant in determining the final performance measure now a more in-depth assessment has been carried out. For example purely temporal factors, such as target speed, which were identified during the parameter estimation stage, may not actually affect any of the modules if none of them makes use of temporal information. Also certain quality measures may not be quantifiable. In this case it may be possible to eliminate the need to measure them, either by using proxy measures or by combining modules to eliminate the need to determine intermediate measures. An example of this might be where the need to measure background clutter is overcome by considering two modules together, as in the ladle tracking example in chapter 4.

3.6 Module Level Analysis

It is now necessary to go through each module in turn and estimate the transfer function relating the input quality measures and external parameters affecting that particular module, to the outgoing quality measures.

3.6.1 Analysis of Individual Modules

From the final block diagram of the system, each module can be extracted and examined in detail. The data streams, $x, y$ should now be determined as well as the elements of the vectors describing the input and output quality measures, $Q_x, Q_y$ and the relevant external parameters, $e$. The nature of the quality measures will depend on the particular module and data type. For example they may be the probability of error or the standard deviation of some output parameter. The combined elements of $Q_x$ and $e$, go up to make the input vector to that module, $C_N$, and the outgoing quality measures, $Q_y$, make up the elements of $A_N$ for the module $N$.

For each module, either a transfer function or matrix, $M_N$, relating the input quality measures and external parameters to the output quality measures must be calculated or estimated. This can be derived either analytically or empirically.
Figure 3-6 A single module in the system. The relationship between the external performance parameters and incoming and outgoing quality measures must be determined.

3.6.2 Analytical Analysis

An analytical analysis is usually preferable where the operating conditions and the action of the module can be modelled mathematically. It has the advantage of giving the developer a clearer understanding of how the performance is being affected by the different input parameters, and how the performance can be improved by tuning the algorithm, and possibly the operating conditions, to move to a more effective operating point. It also does not require any code to be written because the mathematical algorithm, as opposed to its computer code counterpart, should be sufficient to perform the analysis.

There is no general prescriptive method for developing the analytical transfer function for the performance stream from the mathematical algorithm. However, the examples in following chapters give an indication of how it can be carried out for certain types of problem.

3.6.3 Empirical Modelling

If the algorithm and operating conditions are less amenable to modelling, an empirical model of the module is required. This requires an implementation of the module to be modelled, and in some cases an implementation of preceding modules in order to generate the necessary input data. However it does require less mathematical analysis than a purely analytical equivalent. Once an implementation of the algorithm has been written (it does not have to be fully functional, as long as it can be made to perform the processing on the data stream one is interested in), an estimate must be
made of the range of the operating conditions in terms of the input quality parameters and external parameters.

The intention is to generate the performance envelope of the module by sampling the input parameter space, applying those conditions to the algorithm and measuring the performance or output quality parameters under these conditions. The range of the input conditions can be estimated from video footage and measurements taken in situ, or from information from the end user. The resolution of the sampling of the input space must be a trade-off between computational requirements and accuracy. It is recommended that the most detrimental and most commonly occurring parameters are analysed first, as these can often give a rough indication of likely performance at an early stage in the analysis. Hence it can be determined as soon as possible if the system’s performance is going to be close to meeting the specifications.

3.7 Estimation of Final Performance

The final stage is to predict the performance of the system given a particular problem. This requires the measurement or estimate of the probability distribution, \( P(C) \), of the values of the different external parameters, \( C \). The simplest, though possibly most time-consuming, is to generate probability distributions based on sample video footage. If the footage is representative of the conditions under which the system will be operating, a frequency distribution of the different parameter values can be generated by measuring the values of the parameters from images captured at intervals from the footage. This is often sufficient to cover the effects of frequently occurring parameter values. However it may have to be supplemented by estimates of the frequency of less common, though still significant, operating conditions.

Once a probability distribution, \( P(C) \), has been generated, it should then be straightforward to calculate the corresponding probability distribution of the performance, \( P(A) \), using the transfer functions derived in the analysis and \( P(A) = MC \), linking the input parameters and the output performance.
3.8 Framework for General Algorithm Module Analysis

The preceding methodology has been aimed primarily at evaluating specific solutions to specific problems. However, by breaking the system down into modules, it becomes possible to produce more general performance data. If each individual algorithm is treated as a building block from which vision systems can be created, and each algorithm is accompanied by appropriate performance data, then the performance analysis process can be speeded up considerably.

The ultimate aim is therefore to create a framework under which algorithms are evaluated and the results published. The results should be in a form in which the performance data can be re-used.

3.8.1 Universal, Compatible Inputs and Metrics

The key to achieving reusable performance data is to characterise the data using universal performance measures and input parameters. In this problem-specific analysis, only performance metrics that are appropriate to the task are used. Also, the parameters against which performance is evaluated are specific to the problem. For every different type of algorithm, a set of metrics should be developed and agreed upon, which cover as much as possible of the algorithm’s possible output errors or characteristics. If universal input parameters for each algorithm are determined, then the user can select which performance metrics and inputs are relevant to his or her problem. They can then measure the values of the input parameter distribution, $P(C)$, and use the appropriate published data for $M$ to calculate $P(A)$. This can then be repeated for all the algorithms which form the system, and the performance propagated through to give a value for probable system performance with considerably less effort than is currently required.

For example, a line-finding algorithm might have the following full set of performance metrics: probability of finding spurious lines, probability of mis-detecting lines, error in $x$ location, error in $y$ location, error in orientation and error in line length. It may have input metrics: line strength (grey levels), noise level, number of pixels in line in image, length of line in image, strength of none-line pixels, frequency spectrum of none-line pixels and possibly some others. If a full set of data relating the values of the performance metrics to the full input parameters space were available, then the user would be able to measure the probability distribution of the relevant input parameters,
select the appropriate performance metrics for the problem and read off the likely performance.

3.9 Summary

This chapter has described the steps involved in using this methodology to characterise a vision system. It has demonstrated how the effects of the operating conditions can be de-coupled from the system performance. It has shown how the external parameters affecting the performance of the system can be determined, and then shown how the vision system can be broken down into modules to simplify the analysis. The use of quality parameters was then described to predict the overall performance through the system. Finally it was shown how these techniques could be extended so that universal performance data could be used to facilitate the analysis.
Chapter 4: Ladle Tracking

This chapter describes the analysis of the first of the four industrial IP tasks that have been evaluated. It uses a simple vision system for tracking batches of steel around a steelworks. The problem fits into category 4 (as described in section 3.4.2), because there are many unquantifiable parameters, and the performance evaluation is mainly empirical. The analysis shows that under the operating conditions present in the plant, the system will not function with sufficient reliability, and the development of such a system was therefore abandoned in favour of the approach adopted in chapter 5.

4.1 Outline of Problem

In a steelworks where batch production of steel is carried out, it is necessary to transport ladles containing 350 tonnes of molten steel around the plant using gantry cranes and mobile cradles. This batch production method requires that several ladles are in the plant at any one time. These may be at any stage in the production cycle or in the maintenance bay for repair of the heat resistant lining. The site operator must track the position of these ladles in the plant at all times for the following reasons:

- The ladles undergo different processing sequences and their contents are not identical. They should not therefore be mis-identified during transportation around the plant.
- Small amounts of trace elements are left in the ladles after use. These will be important in the production of one grade of steel, but will result in undesirable properties if present in other grades. It is essential therefore that empty ladles are not wrongly identified, as trace elements in the bottom of the ladles could contaminate the different batches.
- The sheer size of the ladles gives them a very high heat capacity. Knowing how long each one has cooled enables the site operator to calculate the amount of heat needed to attain the correct temperature when they are next used. Energy savings are also possible if the scheduling can be organised to minimise the cooling effect of the ladles on the steel, by using those which are still hot from another batch.
The ladles are currently tracked around the plant by human operators entering the ladles positions into the computerised scheduling system as the ladles are moved. However this technique is prone to operator error. The design brief was therefore for a vision system which could track the ladles around the plant and interface with the existing scheduling system. Vision based tracking systems are being considered because the environment is so hostile that other types of tracking system (e.g. transponders) are very prone to failure. The imaging environment is also extremely hostile, as steam from furnaces, dust, wide variations in lighting conditions etc., all have to be contended with. A typical TV image taken in the steelworks is shown in Figure 4-1.

![Figure 4-1 Input video image of ladle being transported by crane around Llanwern Steelworks. The ladle can be seen in the bottom right hand corner of the image (image courtesy of British Steel Corporation).](image)

4.1.1 Image Processing System

There are several ways in which tracking algorithms can be implemented: for a model-based example see [91]. Other techniques, for example snakes [92, 93], have also been used. However for this analysis, a tracking algorithm based on template matching of edge filtered images was chosen, as it was simple and reasonably tolerant of lighting variations. The model of the algorithm assumes that an initial ladle position has been given. The input images are edge-filtered using a Sobel operator, although other operators could have been used (the performance of the edge-filtering operation is not being considered here). The algorithm then performs a template matching operation on the area of the scene in which the ladle could have moved since the last operation (referred to as the search space). It uses templates of the ladle which include all possible
scale and orientation changes since the last operation. The position of the ladle is assumed to be the point in the search space with the highest correlation score, using an exhaustive search maximum finder on a normalised cross correlation measure. The search space is then updated on the simple assumption that the search space for the next operation centres on the previous ladle position. A more sophisticated Kalman-filter based predictive tracking system, as described in chapter 2, could be used. However given the nature of the ladle motion (many sudden changes in acceleration) the conditions are not ideal for a Kalman filter, and its use would complicate the performance analysis. The block diagram showing the operation of the algorithm is shown in Figure 4-2.

![Block diagram of a simple tracking algorithm for use in tracking steel ladles.](image)

4.1.1.1 Edge Filter

The edge filter is a simple Sobel operator which is correlated with the image to produce a digitised edge image, according to

\[ f(m, n) = \sqrt{f_v^2 + f_h^2} \]

\[ \text{eq}(4-1) \]

where \( m, n \) are the pixel co-ordinates in the image. The vertical part of the operator is

\[ f_v(m, n) = \sum_{i=m}^{m+f-1} \sum_{j=n}^{n+f-1} e_v(i, j)s(i - m, j - n) \]

\[ \text{eq}(4-2) \]

and the horizontal part is

\[ f_h(m, n) = \sum_{i=m}^{m+f-1} \sum_{j=n}^{n+f-1} e_h(i, j)s(i - m, j - n) \]

\[ \text{eq}(4-3) \]
where \(i,j\) are the co-ordinates in the convolution masks, which are the Sobel edge filters \(e_v\) and \(e_h\). \(s\) is the digitised scene.

\[
e_h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad e_v = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}
\]

\tag{4-4}, \tag{4-5}

### 4.1.1.2 Template Matcher

After the edge filtering, the template matcher correlates the filtered image \(f(m,n)\) with \(L\) different templates of the expected edge image of the ladle, \(t_l(i,j)\). The ladle, as it appears in the image, is assumed to be within a certain range of the size and rotation of the ladle appearing in the previous image (the ladle is pivoted around the trunion, on the front of the ladle as it appears in figure 4-1, in order to empty the molten steel. Thus templates generated with the ladle rotated around the axis directed into the camera's field of view must also be used). The \(L\) templates are chosen from a library of templates to have a size and rotation within the expected range. Each of the \(L\) templates, \(t_0(i,j), t_1(i,j) \ldots t_{L-1}(i,j)\), is then used to generate a correlation surface, \(g(m,n,l)\), corresponding to the \(l^{th}\) template, calculated according to the cross-correlation measure:

\[
g(m,n,l) = \frac{\sum_{i=m}^{m+L-1} \sum_{j=n}^{n+L-1} t_l(i,j) f(i-m, j-n)}{\sqrt{\sum_i \sum_j |t_l(i,j)|^2 \sum_i \sum_j |f(i,j)|^2}}
\]

\tag{4-6}

### 4.1.2 Maximum Finder

The maximum finder exhaustively scans each of the correlation surfaces, \((l=0, l=1 \ldots l=L-1)\) and finds the maximum value within a search area \(S\), corresponding in \(m,n\) space to the region in the image to which the ladle could have moved since the previous frame. The position of the ladle, \(x,y\), is then taken as the position of this maximum, and the value of \(l\) used to estimate the distance, \(z\), of the ladle from the camera. It is also used to determine which templates will be used in the succeeding matching operation.
4.2 Problem Analysis

4.2.1 Parameters and Ranking

After visiting the steel plant and examining videotapes, all the factors which might affect the system's performance were noted. These parameters were listed and ranked as shown in Table 4-1. The second column ranks these parameters in terms of how severely they are likely to affect the performance (1 = most severe, 12 = least severe). This ranking was performed subjectively by estimating the parameter's effect on how difficult it was for a human to locate the ladle. The third column ranks them in terms of how frequently they are likely to have a value which will be deleterious to performance (1 = most frequent). This was performed by estimating how often each parameter had deleterious effect in a 3 hour sample videotape from the plant.

<table>
<thead>
<tr>
<th>Image Parameter</th>
<th>Rank severity</th>
<th>Rank occurrence</th>
<th>Effect on Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>prior information about ladle position</td>
<td>1</td>
<td>13</td>
<td>none</td>
</tr>
<tr>
<td>lighting levels</td>
<td>2</td>
<td>3</td>
<td>ladle contrast</td>
</tr>
<tr>
<td>flare up from molten steel</td>
<td>3</td>
<td>4</td>
<td>affects camera AGC</td>
</tr>
<tr>
<td>distance from cameras to target</td>
<td>4</td>
<td>10</td>
<td>size of ladle in image</td>
</tr>
<tr>
<td>degree of obscuration due to other objects</td>
<td>5</td>
<td>9</td>
<td>reduces ladle visibility</td>
</tr>
<tr>
<td>degree of obscuration due to steam, smoke, dust etc.</td>
<td>6</td>
<td>7</td>
<td>reduces ladle visibility</td>
</tr>
<tr>
<td>amount of and type of background clutter</td>
<td>7</td>
<td>8</td>
<td>distinguishability of ladle from surroundings</td>
</tr>
<tr>
<td>number and recognisability of boundary features on ladle</td>
<td>8</td>
<td>11</td>
<td>distinguishability of ladle from surroundings</td>
</tr>
<tr>
<td>speed of ladle motion</td>
<td>9</td>
<td>12</td>
<td>position of ladle relative to previous image</td>
</tr>
<tr>
<td>detector noise</td>
<td>10</td>
<td>1</td>
<td>gaussian noise in image</td>
</tr>
<tr>
<td>digitisation noise etc.</td>
<td>11</td>
<td>2</td>
<td>noise in image</td>
</tr>
<tr>
<td>dirt on the lens</td>
<td>12</td>
<td>5</td>
<td>noise, obscuration</td>
</tr>
</tbody>
</table>
Table 4-1 The external parameters affecting the performance of the ladle tracker. The parameters have been ranked according to their estimated severity and their probability of occurring with a value which will be detrimental to the algorithm performance.

Crude numerical estimates had to be made for many of these factors, by watching sample video from the plant, in order to assess their potential seriousness and probability of occurrence. For example, the detector noise was always present, giving it a very high probability of occurring in the sequence. However, the noise level was actually quite low, of the order of 3 grey levels, giving it only a slight chance of being detrimental to the system performance.

From these two lists the effect of the following parameters was expected to be small and they were therefore neglected for the rest of the analysis: detector noise, digitisation noise. Although prior information about ladle position is important when the system is turned on (it is difficult to acquire an initial ‘lock on’ to a ladle) it has no effect during normal operation. For the purpose of this analysis therefore, it is assumed that the initial ladle position is known. Nine parameters remained, of which two could be quantified (distance to camera and speed of ladle). The other seven were not readily quantifiable.

4.2.2 Performance Metric

As the task is to track ladles, the performance measure is how often the system ‘loses’ a ladle, i.e. how often it mis-identifies the background as the ladle. For the purposes of this analysis, it is assumed that once the algorithm has mis-identified the ladle, it does not reacquire the target.

4.2.3 Classification

This problem is one in which, due to the number of unquantifiable parameters, the overall performance can not easily be assessed using rigorous mathematical models. The problem falls into category four; its performance can be estimated, but only by using some empirical testing and dealing with the most serious factors first.
4.3 System Analysis

The system consists of three modules; Module 1 is the edge filter, Module 2 the template matcher and Module 3 the maximum finder.

4.3.1 Interaction of Parameters and Quality Measures

The external parameters interact with the modules at the following stages:

Module 1: Edge Filter. The size of the ladle in the image is dependent upon the distance from the ladle to the camera. The edge and clutter strength is dependent upon all the other image parameters, namely illumination, background clutter, noise, flare, obscuration, ladle features.

Module 2: Template Matcher. The correlation signal depth in the correlation surface is dependent upon the strength of the edges of the ladle in the edge image, the size of the ladle in the image and the strength and shape of the background clutter edges.

Module 3: Maximum Finder. The probability of the maximum finder finding the correct point in the search space is affected by the size of the search space and the height of the maximum in the correlation space relative to the correlation score for the non-target parts of the scene. The height of the maximum relative to the standard deviation of the correlation score for the non-target is referred to here as the correlation signal depth.

The interaction of all these parameters is shown in Figure 4-3.
Figure 4-3 The final performance block diagram. The propagation parameters, namely the edge strength of the ladle image, the background clutter spectrum and the correlation signal depth from template matching, have been added. These are the full set of factors which will be used for the performance analysis.

4.3.2 Simplification of Analysis

Several of the parameters that affect the edge filter's performance, and the nature of the edge clutter, are very difficult to quantify. It is therefore only feasible in this case to take an empirical approach to the analysis. This can be simplified by considering the action of the edge filter and template matcher as a single module. This avoids the need to measure the clutter edges independently of their effect on the template matching algorithm. This simplifies the block diagram to two modules, as shown in Figure 4-4. We can also neglect the feedback system for estimating the search space for the next operation, and assume that the search space for the next matching operation is centred on the previous ladle position. This assumes that we are not using any form of motion prediction, such as a Kalman filter, and that the ladle position has a uniform probability of being anywhere within the search space, S.

Figure 4-4 Simplified block diagram of the ladle tracking system. The edge filter and template matcher have been combined to avoid the need to generate measures of the clutter strength.
4.4 Analysis of Modules

4.4.1 Modules 1 and 2: Edge Filter and Template Matching

The first two modules combined generate the correlation surface from the input image. The quality parameter that measures their performance is the correlation signal depth, $D$, which is dependent upon all the parameters describing the image quality.

![Diagram of Edge Filter and Template Matcher]

Since at least three of the factors: background clutter, ladle recognisability and illumination, are both unquantifiable and extremely difficult to model, an empirical test was carried out. The ladle edge contrast, $\alpha$, defined as the mean difference in grey level between the ladle and the background across the ladle edge, was taken as a proxy measure for illumination. It could be determined from video images of the ladle in the steelworks, and does not require lighting position, shadows etc. to be modelled explicitly. A series of test images of the ladles was taken at the steelworks, at approximately the same distance from the camera, with no obscuration, and under varying illumination conditions. Template images were then generated of the ladle edge image for each size and viewpoint. These templates were then correlated against the test images to generate sample correlation maps.
Figure 4-6 Edge filtered image $f(m,n)$ produced using the Sobel operator. Most of the lines are background clutter. The outlines of the ladle and gantry crane are just visible near the right hand side of the image.

Figure 4-7 Correlation surface generated by correlating the ladle template in the figure with the edge filtered image. The image is scaled with black representing the maximum correlation score. The position of the ladle in the image corresponds to the dark cross near the bottom right hand corner.

The correlation score at the true ladle position, $c_{true\_pos}$, is then determined by visually identifying the ladle position. A frequency histogram of the correlation surface $c(m,n)$ is generated and the correlation surface depth, $D$, is then calculated by measuring the difference between the mean correlation score, $\bar{c}$, and the correlation value at the true position $c_{true\_pos}$, and dividing it by the standard deviation of $c(m,n)$, $\sigma_c$.

$$D = \frac{\bar{c} - c_{true\_pos}}{\sigma_c}$$

eq(4-7)
The correlation depth, \( D \), is then plotted against ladle edge contrast, \( \alpha \), yielding the graph shown in Figure 4-8.

![Graph of correlation signal depth versus ladle edge contrast](image)

**Figure 4-8** Graph of correlation signal depth versus ladle edge contrast.

Although the data are scattered, there is an approximately linear relationship between the contrast and the correlation depth at least above an edge strength, \( \alpha \), of 15 grey levels. Below this value, electronic noise effects start to dominate the measurements of the edge strength. However, the trend enables the correlation depth from the module to be estimated, if the contrast level is known. At this stage, no other parameters, such as obscuration, were analysed.

4.4.2 Module 3: Maximum Finder

The data stream and performance propagation stream for the maximum finder are shown in Figure 4-9.
Figure 4-9 Module 3: The maximum finder. The performance transfer function to be calculated has the correlation signal depth as input and the probability of a tracking error as output.

The probability that a tracking error occurs is the probability that some location in the search space has a better (higher) correlation score than the score at the true ladle position, $c_{true\_pos}$, i.e.

$$P\{c_{\text{max}} \leq D\} = G(D)$$

\text{eq}(4-8)

where

$$D = \frac{c - c_{true\_pos}}{\sigma_c}$$

\text{eq}(4-9)

and

$$c_{\text{max}} = \max(c_1, c_2, \ldots, c_n)$$

\text{eq}(4-10)

$c_1, c_2, \ldots, c_n$ are the random, normalised correlation values from the non-ladle background of $f(m,n)$ (away from the correlation maximum), $D$ is the normalised correlation score at the true ladle position, $c_{true\_pos}$ is the correlation score at the true
ladle position before normalisation, \( c \) and \( \sigma_c \) are the mean and standard deviation of (\( c_1, c_2, \ldots, c_n \)). \( G(D) \) is some function to be determined, which describes the probability of error in one matching operation.

Given the assumption that \( c_1, c_2, \ldots, c_n \) are independent and identically distributed random variables, and that their probability distribution has a negative first differential at the positive tail (true of any distribution which decays at the limit), extreme value theory [53] predicts that at the limit, \( G(D) \), will tend asymptotically to the Gumbel, or Type I extreme value distribution, given by

\[
G(D) \rightarrow \exp(-\exp(-D))
\]

eq(4-11)

even if the variables \( c_1, c_2, \ldots, c_n \) do not have a Gaussian distribution. (This is similar to the Central Limit Theory, which implies an asymptotic normal distribution for the sum of many independent, identically distributed random variables. [53])

This gives the probability of error in one matching operation, and is plotted in Figure 4-10.

![Figure 4-10 Probability of error, G(D), in any one matching operation as a function of correlation depth, D.](image)

The ultimate performance indicator is errors/time, and so the search space, \( S \), should be considered in terms of matching processes/time. It is thus a product of the number of different positions in the image over which the ladle must be searched for, \( A \); the number of possible templates, \( L \); and the frequency with which this operation must be repeated, \( f \); thus
The area $A$, and number of templates, $L$, can be calculated readily from the camera geometry and the maximum ladle speed. Conservative estimates of these values gave the number of template matches per unit time required as approximately $10 s^{-1}$. The overall probability of error per unit time, $R$, is thus

$$R = 1 - (1 - G(D))^S$$

where $G(D)$ is the probability of error in one matching operation, i.e. the probability of the true ladle position not having the highest correlation score.

### 4.5 Estimate of Performance

From sample video footage collected over four days, the probability distribution of the edge strength $P(\alpha)$, with zero obscuration (see Figure 4-11) yielded an empirical distribution for $P(\alpha)$, where $\alpha$ is the one dimensional measure of edge strength in grey levels:

![Edge Strength Distribution](image)

**Figure 4-11** Frequency distribution of ladle edge strength, $P(\alpha)$. Although the median edge strength is around 20 grey levels in an 8-bit image, over 20% of the ladle images had an average step edge strength of less than 5 grey levels.

Combining $P(\alpha)$ from Figure 4-11 with the relationship between $\alpha$ and $D$ shown in Figure 4-8 gives an expected value of the signal depth, $D$, of 4.9. This yields an estimated probability of a tracking error in any one matching operation of 0.007, or 0.07 in one second, giving an average of 14.3 seconds between failures. This is on images
with zero obscuration of the ladle. It is reasonable to assume that including obscuration would reduce performance further. From the point of view of the operator this error rate is completely unacceptable, as the current human operator based tracking system achieves far more reliable performance, of the order of several weeks between errors. Consequently the investigation into a ladle tracking algorithm was abandoned, and a new approach was developed, based on identifying codes fixed to the ladles. The analysis of the ladle identification system is described in chapter 5.
Chapter 5: Ladle Identification

5.1 Background

As was shown in Chapter 4, the expected error rate for a continuous ladle tracking algorithm, designed to replace the current human operator based tracking technique outlined in section 4.1, was unacceptably high. It was proposed that it might be preferable to try and identify the ladles at certain locations around the production cycle. If the timing and identity of a ladle entering any of the processes on the production cycle could be logged, then the ladle’s movement around the cycle could then be reconstructed from these ladle positions and times. This could be possible if a camera were placed at the entrance to each of the different process plants, and the ladles each given a panel with a unique identification code. The camera and image processing system must then be able to identify the ladle by recognising its identification code, and pass the information to the scheduling system. This chapter describes a design for a possible tracking system based on error tolerant codes, and uses an analytical model to predict its performance under a variety of operating conditions.

5.2 Operation

The situation under which the ladle identification system would be expected to operate is as follows:

The cameras will be placed at the entrance or exit to various parts of the process plant, in areas with reasonably well controlled lighting, such that the illumination variation across the panel containing the code is small. The ladles will pass through the camera’s field of view as they go through the various plant processes. The approximate time when a ladle is in the field of view will be known \textit{a priori} from the process control signals. The orientation of the ladle will be fixed and the locus of the ladle position will also be known to within a reasonable degree. The image processing task is thus to locate the ladles’ identification codes and distinguish which of the 25 different ladles is in the field of view.
5.2.1 The Codes

The ladles currently have numbers painted on the sides for manual identification. It is proposed that for the purpose of tracking, an identification code of some sort is placed on the end of the ladle trunion. (The trunion is the protruding lug on the side of the ladle.) This is away from the lip of the ladle, and hence is less likely to be affected by drips of molten steel. The area is round and approximately 600mm in diameter. The code would be manufactured from flame cut steel sheet and would sit in relief on the end of the trunion. The area in relief would be painted white, to maximise the contrast against the dark background. Several different types of codes are proposed and evaluated in this chapter. However they all have fairly similar characteristics: They are all white shapes on a black background, and they all consist of two parts, a location pattern and identification pattern. The location pattern is the same for all identification codes of that type, and has approximately equal areas of black and white. This enables the image processing system to locate the position and size of the code in the field of view. The identification pattern is then unique to each ladle and is in a known position relative to the location pattern. The identification pattern must then be recognised in order to identify the ladle.

5.3 The Proposed Image Processing Scheme

The image processing scheme is fairly straightforward. The system must find the location pattern, extract the identification code from the identification pattern and then decode it to distinguish which ladle is in the scene. It is assumed for the purpose of this study that there will only ever be one ladle in the scene at one time, although the analysis can be readily extended to multiple ladles.

A template matching algorithm is used for finding the location pattern. This will scan over the area where the pattern is expected to be, using different sized templates corresponding to the possible range of distances which the pattern could be from the camera. The location of the pattern is then taken to be:
Using the normalised cross-correlation coefficient where \( m, n \) is the location pattern position, \( s(i, j) \) is the scene and \( (t_1, t_2, \ldots, t_L) \) are the different sized templates of the location pattern with outer dimensions \( I, J \).

Having located the pattern, the identification code must now be extracted from the identification pattern. If numbers are used as the identification pattern then this can be performed using template matching again, with templates of the digits we expect to see. However in all other cases the identification patterns are binary coded with different spatial geometries. It is therefore necessary to classify each of the binary pattern regions as black ‘0’ or white ‘1’. Here the algorithm scans the whole pattern and generates an intensity histogram. This will normally be bi-modal, with one of the peaks corresponding to the black regions of the pattern and one to the white regions. A threshold is now selected as the mean of the intensity distribution.

![Bimodal distribution of the pixel intensity from the two-tone pattern image](image)

**Figure 5-1** Bimodal distribution of the pixel intensity from the two-tone pattern image

All the pixels in the image with intensity below this threshold are now classified as ‘0’ and all those above as ‘1’. Each bit region is then classified as either ‘1’ or ‘0’ according to whether the majority of the pixels in that region are ‘1’ or ‘0’. The
extracted bits are now interpreted according to the particular coding scheme used to generate the identification pattern. This may be a straightforward binary representation of the ladle number, or it may be a more sophisticated error correcting code. The operation of error-correcting codes was discussed in chapter 2 and appendix A, and some of the codes which were described are analysed further on in this chapter.

5.4  Problem Level Analysis

5.4.1  Estimation of Parameters Affecting Performance

After taking some sample video images, observing the sites where the system will be set up, and after extensive discussions with the engineers from British Steel, the parameters affecting the operation of the system were estimated to be as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rank Occurrence</th>
<th>Rank Severity</th>
<th>Description</th>
<th>Likely effect on image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electronic noise</td>
<td>=1 10</td>
<td></td>
<td>Noise in camera and electronics</td>
<td>Shot noise causes small random variations in recorded pixel values. Not likely to be of significant effect compared to other parameters.</td>
</tr>
<tr>
<td>Background Clutter</td>
<td>3 9</td>
<td></td>
<td>Non-pattern structure in image</td>
<td>Could cause other regions of the image to resemble the pattern</td>
</tr>
<tr>
<td>De-gassing dust</td>
<td>6 3</td>
<td></td>
<td>De-gassing process coats ladle in a layer of light coloured dust</td>
<td>All dark areas on pattern will appear much lighter, reducing contrast of pattern.</td>
</tr>
<tr>
<td>Paint damage</td>
<td>8 5</td>
<td></td>
<td>Paint on pattern is damaged during ladle use</td>
<td>White areas of the pattern appear dark. The effect is likely to be more pronounced on the edge of the pattern which is more susceptible to physical damage. Distribution elsewhere is likely to be random</td>
</tr>
<tr>
<td>Steel spillage</td>
<td>9 6</td>
<td></td>
<td>Molten steel drips over end of trunion</td>
<td>White areas of the pattern appear dark. Most likely distribution is vertical stripes obscured.</td>
</tr>
<tr>
<td>Obscuration by plant</td>
<td>7 2</td>
<td></td>
<td>Other plant e.g. cranes pass in front of target</td>
<td>Area of pattern not visible.</td>
</tr>
<tr>
<td>Obscuration by steam etc.</td>
<td>5 4</td>
<td></td>
<td>Smoke and steam from plant passes between camera and pattern</td>
<td>Contrast reduced randomly across image. Temporal nature may permit contrast to be improved if a second image can be captured.</td>
</tr>
<tr>
<td>Flare-up</td>
<td>4 7</td>
<td></td>
<td>Flare from production process causes bright regions in camera field of view</td>
<td>Contrast reduced as camera AGC compensates for bright flares elsewhere in field of view</td>
</tr>
<tr>
<td>Lighting failure</td>
<td>10 8</td>
<td></td>
<td>Controlled lighting fails.</td>
<td>Pattern contrast reduced – possible increase in illumination gradient.</td>
</tr>
<tr>
<td>Pattern appearance</td>
<td>=1 1</td>
<td></td>
<td>Pattern design and distance from camera</td>
<td>Affects the size of the different parts of the pattern and their robustness to errors</td>
</tr>
</tbody>
</table>

87
Table 5-1 Relative ranking of phenomena which might affect the performance of the proposed image processing system. The second column ranks the phenomena according to probability of occurrence (1 = high). The third column ranks them according to severity in affecting system performance.

None of these factors appear to be negligible, as although background clutter and noise do not rank highly in terms of having a detrimental effect on performance, they are both ever present and may still cause errors.

5.4.2 Classification of Problem

Although some of the factors, such as illumination and background clutter do not appear readily quantifiable, they may be approximated by quantitative measures of contrast and empirical measures of background correlation response or background power spectral density. Those factors which involve forms of obscuration and paint flaking can be characterised by a measure of the proportion of the pattern which is obscured or removed, and of the grey level of the obscuring or revealed object. Smoke or steam effects can be approximated by a measure of their influence on the contrast and obscuration. The other factors can all be quantified. The problem is category 2: a problem whose performance characteristics can be mostly determined analytically, with a few approximations of the less quantifiable effects.

5.4.3 User Performance Metrics

As in Chapter 4, the user performance metric of the overall scheduling system will represent a combination of the saving in energy consumption, the reduced number of errors in scheduling and the improvement in the reliability and timing of ladle lining replacement. In terms of the image processing system, this corresponds simply to the probability of a ladle being mis-identified.

5.5 System Level Analysis

5.5.1 Modularisation

With the exception of when codes based on numeric symbols are used, the image processing system can be broken down into five modules:
Module 1. The Correlator. This is the template matcher which produces the series of \( L \) correlation maps, \( c(m,n,l) \) between the scene, \( s(x,y) \), and the template, \( t_l(x,y) \) and passes them to the next module. \( (m \) and \( n \) are pattern position, \( l \) is the template number relating to template size, \( x \) and \( y \) are the pixel co-ordinates in the two images.)

Module 2. The Maximum Finder. This finds the coordinates of the location of the maximum, \( (x,y) \) in the correlation map \( c(m,n,l) \) and passes them, and the size, \( h \), corresponding to the template, \( t_l(x,y) \) as it appears in the image to the next modules.

Module 3. The Image Thresholder. This finds the mean grey level of the pattern, \( T \), and thresholds the image at that grey level. It then passes a binary image \( i(x,y) \) of the pattern to the bit classifier module.

Module 4. The Bit Classifier. This classifies each of the regions which represent bits of the binary code as either ‘1’ or ‘0’ according to whether the majority of the pixels in that region are black or white in the thresholded image. It passes a bitstream representation, \( x \), of the code, to the decoder.

Module 5. The Decoder. This decodes the binary bit-stream into a number, \( q \), identifying the ladle. It performs any error-correction calculations necessary and passes the ladle number to the scheduling system.

When numeric codes are used, modules 1 and 2 are the same as above, but module 3 is a digit classifier, based on template matching using known digit templates, which sends the ladle number straight to the scheduler.

A block diagram of the modules in the ladle identification system and their information stream is shown below.
Figure 5-2 Modules of the ladle identification system
### 5.5.2 Interaction of Parameters

The parameters which affect the performance of the system can be characterised using the variables introduced in Table 5-2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumination and dust on the ladle</td>
<td>$\alpha =$ contrast between the mean pixel grey level of the black regions and the white regions, $\beta =$ mean pixel grey level of black regions.</td>
</tr>
<tr>
<td>Background Clutter</td>
<td>The standard deviation, $\varepsilon$, of the empirical cross-correlation measure between the background and a code, $\Gamma$.</td>
</tr>
<tr>
<td>Obscuration by other plant, obscuration by steam and smoke, paint flaking off, dust on the ladle drips of molten steel on the ladle</td>
<td>The proportion, $\rho$, of the pattern obscured or now visible as an object of grey level, $c$.</td>
</tr>
<tr>
<td>Noise</td>
<td>The standard deviation, $\sigma$, of the noise on the image of the pattern</td>
</tr>
<tr>
<td>Distance from the camera to the ladle, $d$, and the angular width of the camera field of view, $\phi = 45^\circ$.</td>
<td>These can be modelled as the apparent diameter of the ladle in pixels, $h$. This is approximately $h = \frac{512w}{\phi d}$, assuming the image is 512x512 pixels and the width of the code is $w \approx 0.6m$. This reduces to $h = \frac{312}{\phi d}$.</td>
</tr>
<tr>
<td><strong>A priori</strong> information about ladle position</td>
<td>The size of the search space, $S$, measured in pixel positions, in which the ladle is known to be, and the uncertainty in ladle rotation, $\theta$.</td>
</tr>
<tr>
<td>Pattern characteristics</td>
<td>The apparent size of the ladle, $h$, gives a total apparent area, $A$, which can be considered as two regions. These are the area in pixels $M$, of the location pattern and $N$, of the identification pattern respectively. Therefore $M + N = A = \pi \frac{h^2}{4}$. The two areas can also be considered as their relative proportions of the image size, $\mu$ and $\nu$, where $M = \mu A$ and $N = \nu A$, with $\nu = 1 - \mu$. The other parameters are the number of bit regions in the identification pattern, $n$, their mean size $q$, the number of errors which can be corrected in the code $t$, and the number of regions in the location pattern $\alpha$.</td>
</tr>
</tbody>
</table>

Table 5-2 Modelling the parameters which affect the performance of the system, and the variables used to describe them.

The external parameters affect each of the modules as follows:
Module 1, Correlator: contrast, $\alpha, \beta$; clutter, $\Gamma$, and its standard deviation, $\varepsilon$; obscuration, $\rho, c$; noise s.d., $\sigma$; size of location pattern, $M$, number of regions in the location pattern, $\omega$.

Module 2, Maximum Finder: search space, $S'$; noise, $\sigma$.

Module 3, Thresholder: contrast, $\alpha, \beta$; obscuration, $\rho, c$; noise, $\sigma$; size of whole pattern, $A$.

Module 4, Bit Classifier: obscuration, $\rho$, size and number of bit regions, $q$ and $n$; uncertainty in ladle rotation, $\theta$.

Module 5, Decoder: Choice of coding scheme and bit error correction rate, $t$.

5.5.3 Choice and Propagation of Quality Measures

The quality measures are chosen by considering the measures forwarded by the information stream. If we propagate the quality measure backwards from the user’s performance measure of the probability of a mis-identified ladle, we can thus determine how to characterise the information from each of the modules to estimate the user’s quality measure.

Module 5: The quality measure which affects the decoder’s ability to identify the ladle is the probability of bit mis-classification, $p(b)$, as this will yield the probability that the number of errors is greater than $t$. Also, the probability that the estimated position $x, y$ of the pattern is incorrect (i.e. the template matcher has locked on to the wrong part of the image).

Module 4: The quality measure characterising the information passed to the bit classifier which will affect the accuracy of its output, is the probability of the pixels in the binary image being mis-classified, $p(r)$, and the accuracy with which the position of the pattern is known.

Module 3: The internal quality measure affecting the probability of the thresholder mis-classifying the pixels, is the probability that the estimated position $x, y$ of the pattern is incorrect.

Module 2: The probability of the maximum finder locating the true pattern is dependent upon the value of the correlation map at the true location and the distribution of the values of the correlation map at non-target locations.

92
The whole interaction of the quality measures, information stream and external parameters is illustrated in Figure 5-3:

![Figure 5-3](image)

Figure 5-3 The modules of the ladle identification system. The top set of arrows indicates the information stream; the bottom set the external and quality propagation parameters.

5.6 Performance Characterisation of System

It is now possible to analyse the performance of each of the modules in turn at the algorithm level to determine the transfer function between the input parameter(s), the input quality measure(s) and the output quality measure(s).

5.6.1 Module 1: Correlator

For each different sized location pattern template, the Correlator Module is as follows:
The top arrows show the information stream through the module. The remaining arrows indicate the external parameters which act on this module and the quality propagation parameters which are needed to estimate the performance of the following module. The transfer function between \( s(x,y) \) and \( c(m,n) \) is given by eq(5-2). What is needed is to evaluate the transfer function between the external parameters and the quality propagation parameters; the signal to clutter ratio, \( Q \), and the sharpness of the correlation peak, \( \Delta C/\Delta x \).

In order to calculate the signal to clutter ratio, \( Q \), the value of the correlation map at the true signal position and the standard deviation of the background clutter response, \( \varepsilon \), must be evaluated. The value of the signal depth at the true position can be estimated by modelling the scene as being identical to the template, but with contrast \( \alpha \), pedestal grey level, \( \beta \), zero-mean noise, \( n \), with standard deviation \( \sigma \), and a proportion of the pattern \( \rho \) obscured by an object of grey level \( c \).

The correlation signal \( C \) is (neglecting the subscripts for clarity)

\[
C = \frac{\sum_M t(s + n)}{\sqrt{\sum_M t^2} \sqrt{\sum_M (s + n)^2}}
\]

\textbf{eq}(5-2)

If we now let \( s = \alpha t + \beta \) (so \( s \) is a contrast scaled version of the template), then:
\[ C = \frac{\sum_{M} t(\alpha + \beta + n)}{\sqrt{\sum_{M} t^2} \sqrt{\sum_{M} (\alpha + \beta + n)^2}} \]

\[ \text{eq}(5-3) \]

We now let a proportion, \( \rho \) of the pixels of \( s \) equal a grey level \( c \), plus noise \( n \), with the same standard deviation \( \sigma \), as the rest of the pattern:

\[ C = \frac{\sum_{(1-\rho)M} t(\alpha + \beta + n) + \sum_{\rho M} t(c + n)}{\sqrt{\sum_{M} t^2} \sqrt{(1-\rho)M} (\alpha + \beta + n)^2 + \sum_{\rho M} t(c + n)^2} \]

\[ \text{eq}(5-4) \]

Now note that \( \sum_{M} n^2 = M\sigma^2 \). Also, if we assume that the template has an equal number of pixels of value 1 and 0, then \( \sum_{M} t = \frac{M}{2} \) and \( \sum_{M} t^2 = \frac{M}{2} \).

Substituting these in to eq(5-4) yields a value for \( C \) at the ladle position of

\[ C = \frac{\alpha + \beta + \rho(c - \alpha - \beta)}{\sqrt{\alpha^2 + 2\alpha\beta + 2\beta^2 + 2\sigma^2 + \rho[2c^2 - \alpha^2 - 2\alpha\beta - 2\beta^2]}} \]

\[ \text{eq}(5-5) \]

Away from the ladle position, the value of \( C \) is less easy to evaluate analytically, as this requires a model of the background clutter, \( \Gamma \), which is difficult to calculate accurately. Hence an empirical estimate is required.

A sample location pattern was correlated with a sample video image of a ladle with a test pattern occupying the whole of the trunion end, using the correlation measure described above. This yielded a standard deviation of the response, \( \varepsilon \), for \( M = A \) (the entire region of the pattern occupied by a location pattern), of approximately \( \varepsilon = 0.05 \). Since the background response is the normalised sum of \( M \) random variables, \( b \), with some unknown standard deviation, \( \sigma_b \), it can be shown that
Thus for some value of $M$, $M = \mu A$, the standard deviation, $\lambda$, of the background clutter response for a location pattern occupying some proportion, $\mu$, of the total pattern region is:

$$
\lambda = \frac{\epsilon}{\sqrt{\mu}}
$$

Dividing $C$ by the clutter response $\lambda$, yields

$$
Q = \frac{\sqrt{\mu}(\alpha + \beta + \rho[c - \alpha - \beta])}{\epsilon \sqrt{\alpha^2 + 2\alpha \beta + 2\beta^2 + 2\sigma^2 + \rho[2c^2 - \alpha^2 - 2\alpha \beta - 2\beta^2]}}
$$

which is the desired transfer function between the signal to clutter response $Q$, and the external imaging parameters, $\alpha$, $\beta$, $\epsilon$, $\rho$, $\sigma$, and $c$, and the proportion of the pattern region occupied by the location pattern, $\mu$.

The sharpness of the correlation peak can be estimated by considering what happens to the value of $C$ as the template is displaced in one dimension by some distance $\Delta x$. A cross section through the template and the scene in the region of a white block is shown in Figure 5-5.
Figure 5-5 Cross section through template \( t(x) \), and scene \( s(x) \) with template displaced by \( \Delta x \). The shaded regions contribute to the change in the correlation value \( \Delta C \).

As the values of \( t(x) \) are 1 or 0, the change in the correlation score \( \Delta C \) for a mislocation \( \Delta x \) is the difference in area between the two shaded areas. i.e.

\[
\Delta C = e\Delta x(\alpha + \beta) - e\Delta x\beta = e\Delta x\alpha
\]

\text{eq(5-9)}

where \( e \) is the number of edge pixels in the edges running perpendicular to the direction of the displacement. (The vertical scale in Figure 5-5 has been distorted. In real data \( s(x) \) would not have such small (or non-integer) values).

The value of \( e \) is dependent upon the code being used, but can be approximated if we assume that the location pattern consists of \( \omega \) equally sized regions (it does not, but since we are only concerned with estimating the mean size of the regions, this approximation is valid). This gives each region an area of approximately \( M/\omega \) pixels. These regions would each have of the order of \( 2\sqrt{M/\omega} \) edge pixels in any direction, so \( e \approx 2\sqrt{M/\omega} \). It will be shown later that under realistic values of the imaging parameters, the performance of the algorithm is not very sensitive to the estimate of \( e \), so this simple model is reasonable.
Hence

\[ \Delta C = e \Delta x (\alpha + \beta) - e \Delta x \beta = e \Delta x \alpha = 2 \sqrt{M \omega \Delta x \alpha} \]

and the slope of the correlation surface close to the minima, \( \Delta C/\Delta x \), is

\[ \frac{\Delta C}{\Delta x} = 2 \sqrt{M \omega \alpha} \]

eq(5-11)

5.6.2 Module 2: Maximum Finder

The second module, the maximum finder, has an information stream and a performance stream as shown below:

![Diagram](image)

Figure 5-6 Information and performance parameters for the maximum finder module.

The transfer function between \( c(m,n) \) and \( x,y \) is

\[ x,y = \max_{x=m, y=n}[c(m,n)] \]

eq(5-12)
The maximum finder is assumed to perform an exhaustive search of the space $S$ over which $m, n \in S$.

The performance transfer function can be split into two components. One function relates $Q$ and $S_p$, where $S_p$ is the size or number of elements in $S$, to the probability of finding the incorrect peak in the correlation surface, $p(m)$. The other relates $\Delta C/\Delta x$ and $\sigma$ to the expected local error in the position of the peak $e_{xy}$ (assuming we have found the correct peak).

To estimate $p(m)$, we must estimate the probability that the location pattern is at the position in the search space with the best (maximum) correlation score, $c_{\text{max}}$ i.e.

$$P[c_{\text{max}} \leq T] = G(T)$$

where

$$T = \frac{c - c_{\text{true\_pos}}}{\sigma_c}$$

This is the same as the maximum finder operation in chapter 4, and again extreme value theory predicts:

$$G(T) \rightarrow \exp(-\exp(-T))$$

From module 1, we had already evaluated the normalised correlation depth in the correlation surface as $Q$, so $Q = T$ and

$$G(Q) \rightarrow \exp(-\exp(-Q))$$

This is the probability of getting the incorrect position in any one matching operation. The location error rate is thus

$$p(m) = 1 - (1 - G(Q))^S_p$$
as $S_p$ matching operations must be performed to search the space $S$.

The relationship between $\Delta C/\Delta x$, $\sigma$ and $e_{xy}$ can be analysed by considering the shape of the correlation surface in the region of the maximum.

![Diagram](image)

**Figure 5-7 Cross section through the minimum in the correlation surface showing the effect of discretisation when the minimum is centred on a pixel position**

The discretised correlation maximum is shown with the maximum centred on a discrete pixel position. $c(m,n)$ will have a noise component in the vertical direction with standard deviation $\sigma_c = \sigma/\sqrt{N/2}$ (as the noise, $\sigma$, in the scene is summed over $N/2$ samples, then normalised by $2/N$, to generate $c(m,n)$). Thus the probability that the error on the position, $e_{xy} = 1$ pixel is equal to the probability that $[C-(\Delta C/\Delta x)] + \xi > C$ where $\xi$ is the noise on $c(m,n)$ with standard deviation $\sigma_c$. This is equal to

$$p(e_{xy} = 1) = \phi \left( \frac{\Delta C/\Delta x}{\sigma_c} \right)$$

eq(5-18)

where $\phi(z)$ is the integral of the normal distribution function, given by
\[
\phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-\frac{1}{2}t^2} \, dt = \frac{1}{2} \left(1 + \text{erf}(z)\right)
\]

Realistic estimates of \(\sigma_c\) and \(\Delta C/\Delta x\) give approximate values of \(\Delta C/(\Delta x \times \sigma_c) = 6 \times 10^3\). Thus the value of \(p(e_x = 1)\) is vanishingly small. However if the maximum is centred between two pixels, as shown Figure 5-8, the difference, \(\Delta C\), between the correlation at these two values is small and a much lower \(\sigma\) could result in the error in the maximum position being 1 pixel in any direction.

![Discretised correlation minima not centred on a pixel position](image)

**Figure 5-8 Discretised correlation minima not centred on a pixel position**

Therefore for the remainder of the analysis, if the correct peak is found, the position of the correlation maximum will be assumed to be known with a precision of ± 1 pixel.

5.6.3 Module 3: Image Thresher

At this stage, the position and size of the location pattern in the image has been determined. From this, the position of the bit regions can be determined. It is now necessary to threshold the pattern to classify the pixels as either black or white. The thresher operates in two stages: the first stage determines the threshold value and the
second stage then performs the actual image thresholding. However, since their performance is so closely linked, they are considered here as one module.

![Diagram of the thresholding process](image)

**Figure 5-9 Interaction of parameters with the thresholder module**

We already have a model of how the pattern is represented in the scene. Black pattern regions appear as pixels with grey level $\beta$, white regions as grey level $\alpha+\beta$, a proportion $\rho$ of the image is obscured by an area of grey level $c$, and Gaussian noise is present with standard deviation $\sigma$. If we consider the histogram of the pattern shown in Figure 5-10.

![Histogram of pixel intensity](image)

**Figure 5-10 Trimodal distribution of pixel intensity when a region of the pattern is obscured.**

We would expect typically a trimodal distribution centred on $\beta$, $\alpha+\beta$ and $c$, corresponding to the black, white and occluded regions respectively. They are shown
here with $\beta < c < \alpha + \beta$, though this is not necessarily the case (their relative values do not affect this analysis). The two peaks centring on $\beta$ and $\alpha + \beta$ each have area $N(1-\rho)/2$ and the peak at $c$ will have area $\rho N$. The mean grey level in the image, $A$, is therefore

$$A = \frac{1}{2} [\beta(1-\rho) + (\alpha + \beta)(1-\rho)] + \rho c$$

eq(5-20)

This is now set as the threshold and any pixels with grey level $s \geq A$ are classified as white or '1' and any with $s < A$ as black or '0'. The probability, $p(r)$, that a pixel is misclassified is therefore

$$p(r) = (1-\rho)[p(\beta + \zeta > A) + p(\alpha + \beta + \zeta < A)] + \frac{\rho}{2}$$

eq(5-21)

where $\zeta$ is the zero-mean noise. Rearranging this gives

$$p(r) = (1-\rho)[p(\zeta > \rho(c - \alpha - \beta) + \frac{\alpha}{2}) + p(\zeta < \rho(c - \alpha - \beta) + \frac{\alpha}{2})] + \frac{\rho}{2}$$

eq(5-22)

Assuming $\zeta$ has a Gaussian distribution with standard deviation $\sigma$ yields

$$p(r) = (1-\rho)[\phi\left(\frac{\rho(c - \alpha - \beta) + \frac{\alpha}{2}}{\sigma}\right) + \phi\left(\frac{\rho(c - \alpha - \beta) + \frac{\alpha}{2}}{\sigma}\right)] + \frac{\rho}{2}$$

eq(5-23)

and $\phi(z)$ is given in eq(5-19).

5.6.4 Module 4: Bit Classifier

The bit classifier interacts with the performance stream as follows
From the location and size information from module 2, the bit-classifier knows where in the binary image each of the bit regions are expected to appear. It classifies each bit according to whether the majority of the pixels in the region corresponding to that bit in the binary thresholded image are set to black or white.

Originally in the investigation, it was believed that the ladles may also rock in the crane, causing the pattern to appear with some a random axial rotation, $\theta$, where $-10^\circ < \theta < 10^\circ$. This would introduce synchronisation errors in the code. To overcome this, the toothed code was designed such that it was robust to synch errors using the scheme outlined in appendix A. However, information received later in the investigation suggested that $-1^\circ < \theta < 1^\circ$, and the rotational tolerance was no longer necessary.

A bit region has an area of $q$ pixels, with an uncertainty in its location of $\delta x$ pixels, and each pixel has a probability $p(r)$ of being mis-classified. The effective region over which we can sum the thresholded pixels is $v$, where

$$v = (\sqrt{q} - 2\delta x)^2$$

if the region is approximated as a square and $\sqrt{q} \gg \delta x$, as shown in Figure 5-12. In section 5.6.2, it was shown that $\delta x < 1$. 

Figure 5-11 Parameter interaction with the bit-classifier module.
Figure 5-12 Illustration of the difference between the known region size and the actual region size, given an uncertainty in the region position of $\delta x$.

The probable number of mis-classified pixels is approximately a normal distribution having mean $v p(r)$ and variance $v p(r)(1-p(r))$. Thus the probability, $p(e)$ that more than $v/2$ pixels are mis-classified and hence the bit is mis-classified, is:

$$p(e) = \phi \left( \frac{v/2 - vp(r)}{\sqrt{vp(r)(1 - p(r))}} \right)$$

where $\phi(z)$ is again given in eq(5-19).

5.6.5 Module 5: Decoder

The bit-stream or pattern corresponding to the identification code has been extracted with a known probability of any one bit having been incorrectly classified of $p(e)$. This bit-stream is passed to the decoding routine.
The operation and analysis of the decoder depends on the type of identification code being used. In this section, six different identification codes are proposed, and their likely performance estimated and compared. Each uses an outer circle as the location pattern. The first code is simply an Arabic numeral inside a circular location pattern, with the number defining the ladle. The next two contain a binary encoding of the number using concentric rings in two different formats. The last three all use a form of error-correcting code as described in appendix A.

Since there are 25 ladles, 5 bits of information must be encoded in each of the non-numeric patterns. For the two codes using binary encoding, this means only 5 bits are required. However, for the three codes using error-correcting codes, the number of bits to be encoded depends on the choice of code. For the analysis, Bose-Chaudhuri-Hocqengen (BCH) coding was used to encode $k = 5$ information bits. Using $n$ bits in total, Figure 5-14 gives the maximum error correction rate $t/n$, where $t$ is the number of bit errors that can be corrected.
Figure 5-14 Graph showing the maximum error correction rate versus code length for the codes encoding 5 information bits. The correction rate has a series of peaks at $n = 2^p - 1$, where $p$ is an integer. The maximum correction rate is bounded by the Elias and Plotkin theoretical bounds at around 25% [78].

For each of the three error-correcting pattern designs, a BCH code was chosen of length $n = 2^p - 1$ to maximise the correction rate for 5 information bits.

The six codes are as follows:

1. Numeric

Figure 5-15 Numeric code. Digits 1-25 inside a circle. Circle is used for location and digit indicates ladle identity.
2. Binary Concentric Circles

![Binary Concentric Circles](image)

Figure 5-16 Binary circle code. Outer circle used for location. Five concentric circles encode 5 bit binary number representing the ladle identification number.

3. Binary Concentric Circles with Mirroring.

![Binary Concentric Circles with Mirroring](image)

Figure 5-17 As Figure 5-16 but with the bit represented by the inner ring on the top half of the pattern switched with the outer ring, and so forth.

4. Bar Code with error correction

![Bar Code with error correction](image)

Figure 5-18 Vertical bar code using a BCH code with $n = 15$ stripes in total, maximum of $t = 3$ errors.
5. Toothed Bullseye code with cyclic error correction.

![Figure 5-19 Bullseye code. Centre bullseye and outer ring provide location information. Teeth in outer ring encode coded version of the ladle number using a BCH code with \( n = 31 \) teeth in total, maximum of \( t = 7 \) errors.]

6. Array code with parity bits

![Figure 5-20 Block array code using a BCH code with \( n = 15 \) blocks in total, maximum of \( t = 3 \) errors.]

5.6.6 Empirical Correlation Errors of Numeric Codes

The first numeric code uses the simpler three-stage processing scheme of template matching against the circular location pattern and then again against the numerals. The empirical correlation scores as the numerals 1 to 20 are matched against themselves and each other are shown in Figure 5-21. The scores have been scaled to form a greyscale image. White squares indicate a correlation score of one, black of zero.
Figure 5-21 The empirical correlation scores as the numerals 1 to 20 are matched against themselves with zero noise. White squares indicate a correlation score of one, black of zero.

The white squares on the diagonal represent the correlation maxima which occur when each digit is correlated against itself when there is no noise. However as the ratio of contrast, $\alpha$, to noise, $\sigma$, is reduced, the brightness of the diagonal squares decreases and the off-diagonals increase, increasing the probability of mis-classification. The effects of a lower contrast to noise ratio is shown in Figure 5-22.
Figure 5-22 The empirical correlation scores as the numerals 1 to 20 are matched against themselves with added noise (SNR ≈ 1.4). The squares on the diagonal are no longer white, and some of the off diagonals are brighter. This illustrates the increased probability of incorrect pattern matching in the presence of noise.

The precise effects of the ratio of signal $\alpha$ to noise $\sigma$ are difficult to estimate for the numeric codem, but the probability of a mis-classified digit under zero obscuration seems not to be significant until $\alpha/\sigma$ falls to around 3. However, numeric codes are far more susceptible to the effects of obscuration. The approximate minimum value of $\rho$, for which an 18 could be confused for a 13, or an 8 for a 3 is $\rho = 0.01$. (This is the proportion of the area of the image which is different between numbers 3 and 8, i.e. the left-hand side of the loops.)
5.6.7 Comparison of the Code Designs

The tolerance of the six different codes described above to obscuration, \( \rho \), can be compared as follows. For each, the area of the location pattern, \( M \), was fixed at 37\% of the total pattern size, \( A \) (\( \mu = 0.37 \)). The area of the identification pattern varies depending on the code design, from 63\% of \( A \) (\( \nu = 0.63 \)) for the concentric codes to 27\% for the numeric code. From the area of the identification codes, a mean and minimum bit region size can then be calculated for each of the codes. For the toothed, array and bar codes, these two values are the same, as all the bit regions have the same area. For the circle codes, the minimum region size is the size of the central circle. This then allows estimates to be made of a value for the minimum obscuration which could cause an error \( \rho_{\text{min}} \). This is the worst case, such that the obscuration is the minimum amount, in a position such that in excess of 50\% of more than \( t \) bit regions are mis-classified. The mean tolerance to error can be estimated from the probability, \( p(f) \), that the number of bit regions mis-classified by the previous module is greater than \( t \), where

\[
p(f) = \sum_{x=0}^{n} \frac{n!}{x!(n-x)!} p(e)^x (1 - p(e))^{n-x}
\]

\[\text{eq}(5-26)\]

<table>
<thead>
<tr>
<th>Type of Code</th>
<th>Size of location pattern, ( M )</th>
<th>Size of identification pattern, ( N )</th>
<th>Number of bits, ( n )</th>
<th>Mean bit size, ( q )</th>
<th>Minimum bit size, ( q_{\text{min}} )</th>
<th>Bit error tolerance, ( t )</th>
<th>Minimum obsc. tolerance, ( \rho_{\text{min}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Numeric</td>
<td>0.368</td>
<td>0.267</td>
<td>N/A</td>
<td>0.01</td>
<td>N/A</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td>2. Binary Con. Code</td>
<td>0.368</td>
<td>0.632</td>
<td>5</td>
<td>0.126</td>
<td>0.018</td>
<td>0</td>
<td>0.009</td>
</tr>
<tr>
<td>3. Mirrored Binary Con.Code</td>
<td>0.368</td>
<td>0.632</td>
<td>10</td>
<td>0.063</td>
<td>0.009</td>
<td>0</td>
<td>0.004</td>
</tr>
<tr>
<td>4. Bar Code</td>
<td>0.368</td>
<td>0.401</td>
<td>15</td>
<td>0.027</td>
<td>0.027</td>
<td>3</td>
<td>0.094</td>
</tr>
<tr>
<td>5. Toothed Bullseye Code</td>
<td>0.5</td>
<td>0.5</td>
<td>31</td>
<td>0.016</td>
<td>0.016</td>
<td>7</td>
<td>0.121</td>
</tr>
<tr>
<td>6. Array Code</td>
<td>0.368</td>
<td>0.401</td>
<td>15</td>
<td>0.027</td>
<td>0.027</td>
<td>3</td>
<td>0.094</td>
</tr>
</tbody>
</table>

Table 5-3 Comparison of the parameters for the various code designs. These show the tolerance to obscuration of the different pattern designs. All sizes are expressed here as a proportion of the total pattern size, \( A \).
From Table 5-3, the design with the greatest minimum obscuration tolerance is the toothed bullseye code. The remainder of the calculations are therefore based on the bullseye code.

5.7 Estimation of Final Performance

A full estimate of performance is not possible without cataloguing the probability distribution of the operating conditions, \( P(C) \). However, the two most likely failure modes are a loss of contrast due to de-gassing dust on the pattern, and obscuration by other plant. Estimates will therefore be made of the tolerance to these two factors.

5.7.1 Tolerance to de-gassing dust

If we take measurements of all the other factors during ‘normal’ operating conditions, and measure their likely error rates, then by reducing the contrast, \( \alpha \), to simulate the build up of de-gassing dust, we can estimate the expected error rate as a function of de-gassing dust.

Measurements were taken from video images of the plant, and estimates of the scene geometry yielded the following values for the parameters under ‘normal’ operating conditions (no obscuration, expected normal lighting, no dirt or flare-up). A ladle under these conditions could be expected to look approximately as shown in Figure 5-23, though the field of view may be reduced in the final implementation by using different lenses.

![Figure 5-23 Image of ladle taken from steelworks with toothed bullseye code superimposed on the image.](image)
The video measurements gave an approximate background grey level for the black region of the patterns, \( \beta \), of 130, a contrast, \( \alpha \), of 20 and a noise standard deviation, \( \sigma \), of 5, yielding a contrast/noise ratio of 4. (The figures are summarised in Table 5-4.) Using eq(5-23), eq(5-25) and eq(5-26) with obscuration, \( \rho = 0 \), yields a probability of incorrectly reading the pattern if it is located, \( p(f) \), of less than \( 10^{-16} \), and of failing to locate the pattern, \( p(m) \), of 0.02. Since the overall probability of error, \( p(l) \), is the probability of a failure due to either mode, then

\[
p(l) = p(m) \cup p(f) = p(m) + p(f) - p(m)p(f)
\]

eq(5-27)

Hence under these estimated conditions, the overall probability of error, \( p(l) \), is 2\%, due to the danger of clutter being mistaken for the ladle.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimated Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contrast, ( \alpha )</td>
<td>20 grey levels</td>
</tr>
<tr>
<td>Grey level of ‘black’ regions of pattern, ( \beta )</td>
<td>130</td>
</tr>
<tr>
<td>Noise, ( \sigma )</td>
<td>5 grey levels</td>
</tr>
<tr>
<td>Size of search space, ( S_p )</td>
<td>2500 pixels</td>
</tr>
<tr>
<td>Clutter response, ( \varepsilon )</td>
<td>0.05</td>
</tr>
<tr>
<td>Size of ladle in image, ( A )</td>
<td>1200 pixels</td>
</tr>
</tbody>
</table>

Table 5-4 Estimates of parameter values for the operating conditions in the steelworks

To simulate the effect of de-gassing dust, the grey level of the black regions, \( \beta \), was increased while the contrast, \( \alpha \), was reduced. Thus the appearance of the white regions is held constant, but the black regions are gradually ‘whitened’ until they appear to have the same grey-level as the white regions. The values for each of the intermediate error probabilities; incorrect pattern position, \( p(m) \), misclassification of pixel, \( p(r) \), misclassification of bit region, \( p(e) \), misidentification of ladle, \( p(f) \), and the overall error rate, \( p(l) \), are shown in Figure 5-24 as the contrast/noise ratio is reduced.
Figure 5-24 Values of the various error probabilities at the different stages of processing as a function of contrast/noise ratio, given by equations, eq(5-17,23,25,26 and 27). The different probabilities are: incorrect pattern position, \( p(m) \); misclassification of pixel, \( p(r) \); misclassification of bit region, \( p(e) \); misidentification of ladle, \( p(f) \); and the overall error rate, \( p(I) \). Note that the overall error rate is not necessarily the highest, as errors at the pixel and bit-classification stage are corrected for by subsequent modules.

If the contrast to noise ratio is greater than 1.6, then the error rate is dominated by the performance of the location algorithm which, as can be seen from Figure 5-24, has an error rate staying roughly constant at around 2%. If the contrast to noise ratio is less than 1.6, then the effects of the performance of the identification algorithm become significant. The performance then reduces very rapidly, with a 10% error rate when the contrast to noise ratio falls to 1.2. Using these estimates, the system is tolerant to degassing dust up to a contrast of around 8 grey levels. An image with a contrast/noise ratio of 1.6 is shown in Figure 5-25.
The tolerance to de-gassing dust can be improved drastically by changing the apparent size of the pattern in the image, A. If the camera can be set up such that the ladle occupies 20000 pixels (approximately 13% of the 512×512 image), the performance of the identification algorithm does not begin to fall off until the contrast/noise ratio reaches 0.4. However this would require tighter constraints on the physical size of the search space to avoid deleterious affects on the location algorithm. An image with a contrast to noise ratio of 0.4 is shown in Figure 5-26. However at these contrast levels, any illumination variation across the pattern would dominate the performance of the thresher. This level of contrast drop could only be tolerated if the lighting conditions were to be kept almost completely uniform.
5.7.2 Tolerance to obscuration

Using the same estimates of the parameters for the operating conditions given in Table 5-3, the tolerance of the system to obscuration can now be calculated. If the obscuration is assumed to be randomly positioned in the pattern image, the probability of error at the intermediate stages can be determined as a function of obscuration, \( \rho \), from equations eq(5-17, 23, 25, 26 and 27).

![Figure 5-27 Error probabilities as a function of obscuration, \( \rho \), given by equations, eq(5-17, 23, 25, 26 and 27). The different probabilities are: incorrect pattern position, \( p(m) \); misclassification of pixel, \( p(r) \); misclassification of bit region, \( p(e) \); misidentification of ladle, \( p(f) \); and the overall error rate, \( p(l) \).](image)

Again, at low levels of obscuration (\( \rho < 0.3 \)) the overall error rate is dominated by the performance of the location algorithm and has a value of around 0.02. However, once \( \rho \) exceeds 0.37, the performance of the identification algorithm becomes more significant and its failure rate reaches 10% when \( \rho = 0.375 \). This gives the system a tolerance to random obscuration of around 35%.
5.8 Summary

This analysis has provided not only a numerical estimate of the final performance of the ladle identification system, but also a series of analytical equations defining that performance. These expressions are extremely useful, as they now enable the system and code design to be optimised, by varying the design parameters, such as distance from camera to ladle, relative sizes of the location and identification patterns, choice of error-correcting code etc.

This chapter has developed a framework and analysis to find the design parameters that predict the most reliable performance given the operating conditions and constraints. This allows an optimal solution to be engineered in advance, rather than being chosen arbitrarily, as might otherwise be the case.
Chapter 6:  Electron Microscope Lens Aberration Determination

This chapter describes a model for the prediction of the errors in determining the lens aberrations in a transmission electron microscope (TEM). It analyses a previously described technique [75] based on the cross-correlation of a diffractogram from a TEM with a synthetic diffractogram. It uses a simple algorithm, and although it is presented in the style of the methodology, the main content of the analysis is not in the parameter analysis or modularisation, but in the theoretical models developed used to predict the correlation functions and errors. Some of this work was carried out jointly with my supervisor, Dr M Forshaw: the analyses in sections 6.7.2.2 and 6.7.2.3, and the work in section 6.7.3.1 are the result of an equal collaboration. The remainder of the work presented is my own.

6.1 Operation of a Transmission Electron Microscope (TEM)

Electron microscopes use the interaction of a high-energy beam of electrons with atoms within a specimen to produce very high-resolution images of that specimen. The short electron wavelengths which are achievable enable far higher resolution images to be obtained than is possible using an optical microscope. Several variants of electron microscope exist. We are concerned here only with transmission electron microscopes, or TEMs. A TEM contains an electron gun, which emits a stream of electrons. These are accelerated by a high-potential anode, typically at 100-400kV, and focused by a series of electromagnetic lenses, called the condenser lenses. These control the illumination of the image by varying the width of the beam. The beam is aimed at the specimen, which is a thin slice of material, usually frozen if it is an organic substance. The beam electrons interact with the specimen in one of three ways: inelastic (high angle) deflection due primarily to interaction with the core and valence electrons of a specimen atom, elastic (low angle) deflection due to interaction with the outer electrons in the specimen, and no deflection if it does not interact with the specimen. The varying degrees of interaction contribute to the formation of the TEM image. A further series of lenses beneath the specimen, the imaging lenses, then create the magnified image from the electron beam. This electron beam image is then projected onto a fluorescing surface.
of zinc and cadmium sulphide. This emits visible light when struck by the electrons. A
35mm camera or a CCD camera can then record the image. The whole microscope
operates in a very high vacuum, typically 10^-6 Pa, to prevent interaction of the beam with
non-specimen particles. A fuller description of TEMs is given in [94].

6.2 Image Formation Process in the TEM

The phase image formation process in the TEM can be modelled as follows. The
incident wave which falls on the object plane can be described by:

$$\psi = \psi_0 \exp(2\pi k z)$$

Eq(6-1a)

where $k$ is the wave number, $z$ is displacement along the optical axis and $\psi_0$ is the
amplitude. After interaction with the specimen, the amplitude of the transmitted wave
immediately beyond the specimen may be written in the form:

$$\psi = \psi_0 a_s(r) \exp[i \phi_s(r)] \exp(2\pi k z)$$

Eq(6-1b)

where $a_s(r)$ describes the local absorption and $\psi_s(r)$ describes the local phase shift in the
wave front due to interaction with the specimen. $r$ is the position vector in the specimen
plane.

If the specimen is only weakly absorbing, $a_s(r) \approx 1$, and if the phase shift $\phi_s(r)$
<<1, then the exponential term in equation(6-1b) can be expanded as a Taylor series of
the form:

$$\psi_s(r) \approx 1 + i \phi_s(r) - \varepsilon_s(r) + ...$$

Eq(6-1c)

where $\varepsilon_s(r) = 1 - a_s(r)$. In a perfect imaging system, the wave amplitude in the plane of
the magnified image is given by: (For simplicity the remainder of the analysis shall be
described in one-dimension.)

$$\psi_m(x') \approx 1 + [i \phi_s(x') - \varepsilon_s(x')] \exp(2\pi k z)$$

Eq(6-1d)
The $z$ dependence can be suppressed by appropriate choice of origin. $x'$ is the displacement in the magnified image plane.

### 6.3 Lens Aberrations

In practice the imaging process is not perfect. An accelerating voltage of 1MV results in electrons with a wavelength of 0.00087nm and an ideal resolving power, neglecting the effect of the aperture of the TEM, approximately equal to the wavelength. However the resolution is restricted in practice by aberrations. Some of these aberrations (principally the spherical aberration) are inherent in the imaging process and cannot be corrected for. These restrict the resolving power to around 0.10nm [94]. Other aberrations are caused by manufacturing imperfections and contamination, reducing the resolving power further. The effect of these aberrations is to create a wave which differs from the ideal wave surface. Figure 6.1 shows how the true wave differs from the ideal, coherent spherical wave. The image of any object point that is ideally formed from a converging spherical wave is distorted by aberrations in the lenses.

![Diagram of lens aberrations](image)

**Figure 6-1 Image formation in a lens system: wave aberration and image aberration. (from [77])**

The effect of aberrations can be analysed as follows. Consider a specimen with a single spatial frequency, $q_0$, whose amplitude transmission is described by:

$$\psi_q(x) = 1 - e^{-\xi_q} \cos(2\pi q_0 x) + i\xi_q \cos(2\pi q_0 x)$$

\[eq(6-1c)\]
If this object is illuminated by a simple plane wave, three waves are produced. A primary wave which passes straight through the specimen with no interaction, and two diffracted waves corresponding to the spatial frequencies $\pm q$.

If the imaging system is perfect the waves converge to form an exact magnified image of the form:

$$\psi_m(x') = 1 + (-\varepsilon_q + i\varphi_q \cos(2\pi q_0 x'))$$

Equation 6-1f

However, aberrations in the imaging system cause an additional phase shift, $\exp[-iW(q)]$, which is a function of spatial frequency. Thus the image amplitude now becomes:

$$\psi_m(x') = 1 + (-\varepsilon_q + i\varphi_q \cos(2\pi q_0 x') \exp[-iW(q)]$$

Equation 6-1g

The image intensity is therefore given by:

$$I(x') = |\psi_m(x')|^2$$

$$= 1 - 2 \cos W(q) \varepsilon_q \cos(2\pi q_0 x') + 2 \sin W(q) \varphi_q \cos(2\pi q_0 x) + ...$$

$$= 1 - D(q) \varepsilon_q \cos(2\pi q_0 x') - B(q) \varphi_q \cos(2\pi q_0 x) + ...$$

Equation 6-1h

The purely phase shifting aberration $W(q)$ has generated a variation in the intensity image. The term, $2\cos W(q)$, which is the amplitude contrast transfer function (CTF), is relatively insignificant for weakly absorbing objects, as $\varepsilon_q$ is typically much smaller in magnitude than $\varphi_q$ at all spatial frequencies. The more important term is the phase contrast transfer function, which is given by:

$$B(q) = -2\sin W(q)$$

Equation 6-1i

Hence the image intensity can be approximated by:

$$I(x') = 1 - B(q) \varphi_q \cos(2\pi qx')$$

Equation 6-1j

In principle, aberrations can be corrected for if they can be determined, by adjusting the currents in the lenses. Further corrections can be made with the use of off-
axis stigmator coils. One way of determining the aberrations is by using images of an amorphous carbon film. Such films approximate to ‘white noise’ objects. They diffract the electron beam over a wide range of angles, and the magnitude of \( q \) is approximately independent of spatial frequency. By taking the Fourier transform of the image intensity, it is possible to obtain an ‘image’ of the phase CTF. Considering the effect of the range of spatial frequencies yields an image intensity for the carbon film specimen of:

\[
I(x') = 1 - \sum_{q=0}^{q_{\text{max}}} B(q) \varphi_q \cos(2\pi qx)
\]

where the summation is the result of the wide range of spatial frequencies present in the carbon film. The maximum spatial frequency, \( q_{\text{max}} \), is limited in practice by the slight temporal and spatial incoherence due to the finite size of the source aperture. This results in an envelope function which attenuates the higher spatial frequencies. This is considered further in section 6.9.2.2. Taking the Fourier transform of \( I(x') \) yields:

\[
A(q) \propto \sum_{q_i=0}^{q_{\text{max}}} (\delta(q, q_i) + \delta(q, -q_i)) B(q) \varphi(q)
\]

The two dimensional power spectrum, given by \( AA^* \), can be displayed as an image. This is known as the diffractogram, and has the form:

\[
A(q)A(q)^* \propto \sum_{q_i=0}^{q_{\text{max}}} (\delta(q, q_i) + \delta(q, -q_i)) B(q)^2 \varphi(q)^2
\]

The series of random delta functions result in a white noise image which is modulated by the phase CTF squared, and \( \varphi_q(q) \), which is roughly constant over all spatial frequencies. Thus the pattern formed can be used to determine the lens CTF directly, as the intensity of the image is proportional to product of the square of the CTF modulating function and white noise. This in turn can be used to calculate the aberrations.

The aberration function, \( W \), of an electron beam can be described by a Taylor expansion of the difference between the actual wave surface and the ideal reference sphere. The reference sphere would be the transfer function if no aberrations were present. In polar coordinates,
where $\phi$ is the azimuthal angle and $\theta$ is the (radial) diffraction angle, which is related to the spatial frequency, $\theta = \frac{\lambda}{q}$. The full definition of the different aberration terms ($A_0, A_1 \ldots C_3$) is given in [74], from which the equation is taken. As the diffractogram yields the power spectrum of the phase shift of the aberration function, the odd terms in the expansion in $\theta$ are not apparent in the image. However they can be determined by injecting different degrees of tilt into the electron beam, giving a set of different diffractograms as shown in Figure 6-3. These have apparent aberrations, which in turn relate to the true aberrations, according to:

$$W(\theta, \phi) = |A_0|e^{i\phi} + \frac{1}{2}|A_1|e^{i\phi} + \frac{1}{2}|C_1|e^{i\phi} + \frac{1}{3}|A_2|e^{i\phi} + \frac{1}{3}|C_3|e^{i\phi} +$$

$$+ \frac{1}{4}|B_2|e^{i\phi} + \frac{1}{4}|B_2|e^{i\phi} + \frac{1}{4}|B_2|e^{i\phi} + \frac{1}{4}|B_2|e^{i\phi} + \frac{1}{4}|B_2|e^{i\phi} +$$

\[
\text{eq}(6-1n)
\]

where $\phi$ is the azimuthal angle and $\theta$ is the (radial) diffraction angle, which is related to the spatial frequency, $\theta = \frac{\lambda}{q}$. The full definition of the different aberration terms ($A_0, A_1 \ldots C_3$) is given in [74], from which the equation is taken. As the diffractogram yields the power spectrum of the phase shift of the aberration function, the odd terms in the expansion in $\theta$ are not apparent in the image. However they can be determined by injecting different degrees of tilt into the electron beam, giving a set of different diffractograms as shown in Figure 6-3. These have apparent aberrations, which in turn relate to the true aberrations, according to:

$$A'_0 = A_0 + A_1 \tau + C_1 \tau + A_2 \tau^2 + \frac{1}{3}B_2 \tau^3 + \frac{2}{3}B_2 \tau + C_3 \tau$$

$$A'_1 = A_1 + 2A_2 \tau + \frac{2}{3}B_2 \tau + C_3 \tau$$

$$A'_2 = A_2 + 3C_3 \tau + C_3 \tau$$

$$A'_3 = C_4 + 2C_3 \tau$$

\[
\text{eq}(6-2,3,4,5,6,7)
\]

where $\tau$ indicates the complex angle describing the injected beam tilt through the diffraction plane, and the primes indicate apparent aberrations [74]. The effect of errors in measuring the apparent aberrations on the final estimates of the true aberrations has been described in [95]. For the remainder of this analysis, we shall be considering the results of beam-tilt diffractograms, and attempting to determine only the apparent aberrations. For simplicity the primes will be dropped and further references to the main aberrations, namely defocus $C_1$, first order astigmatism $A_1$ and spherical aberration $C_3$, will refer to their apparent manifestations in the beam tilt images.

In the beam-tilted diffractogram of an object with a flat power spectrum, a wave aberration function $W(\theta, \phi)$ would result in an image with intensity of the form
\[ I(r,\phi) = \sin^2 \left\{ \pi \lambda C_3 r^4 / 2 - \pi \lambda C_1 r^2 / 2 - (\pi \lambda A_1 r^2 / 2) \cos(\phi - \phi_{22}) \right\} \]

\text{eq}(6-8)

where \( r \) is the radial distance in the diffraction plane; \( A_1, C_1 \) and \( C_3 \) are the apparent first order astigmatism, defocus, and spherical aberration, and \( \lambda \) is the electron wavelength. \( \phi_{22} \) is the angle of primary astigmatism. The diffraction pattern only contains intensity information, so the brightness of the recorded image varies as \(|\text{CTF}|^2\). This results in a \( \sin^2 \) transform, indicating the power spectrum of the phase-shifted beam. The expression inside the curly braces \( \{ \} \) is the Seidelian equation, which shows the dependence of each of the types of aberration; defocus, astigmatism and spherical aberration, on spatial frequency \([75]\). Zeroes in the CTF appear as minima in the diffraction pattern. The approaches to determining the aberrations from these images can be broken down into two types, the Diffractogram Tableaux method described in this analysis, and the Beam-tilt/cross correlation (BTXC) method. The latter is performed by cross-correlating the patterns obtained with the beam tilted with the on-axis diffraction pattern. The peak in the correlation maps indicate by how much the images have been displaced and provides the extra input variables to solve for the CTF \([77]\). A description of how this can be carried out in practice is given in \([96]\). The problem we are concerned with here is the diffractogram tableaux method; determining the CTF by fitting the expression in eq(6-8) to the diffractogram image to determine the values of \( A_1, C_1, \phi_{22} \) and \( C_3 \).

Figure 6-2 shows a transmission electron micrograph of an amorphous carbon film with a small crystalline inclusion, and Figure 6-4 shows a set of diffractograms, each obtained by illuminating the specimen from different angles of incidence. The carbon film micrographs were visually almost identical, hence only one is shown in Figure 6-2. Note that the positioning of the diffractograms in Figure 6-4 is intended only to indicate that they were obtained from images with four different tilt angles relative to the central, zero tilt image - the diffractograms are Fourier transform power spectra of the original images. The ‘noisy’ structure of the diffractograms is the power spectrum of the carbon film, while the modulating fringe patterns indicate the CTF of the imaging system. The shape of the fringes indicate the phase-shift of the electron beam due to aberrations in the electron imaging system. The bright spots in the diffractogram are produced by the regular structure of crystalline inclusions in the specimen.
Figure 6-2 The transmission electron micrograph of an amorphous carbon film with a crystalline inclusion in the middle. The image width is approximately 18 nm. (Image courtesy of A. Kirkland)

Figure 6-3 A set of five diffractograms, each obtained by Fourier transforming an image of the same amorphous carbon film specimen shown in Figure 1. The images differed only in the tilt applied to the illuminating beam - nominally zero for the central diffractogram, and approximately ±10 mrad about the x or y axes
for the other four. The diffractograms have been cropped for display purposes and the central maxima have been suppressed. The bright spikes in the image are Fourier components corresponding to the crystalline inclusions. The maximum frequency of each pattern as it appears here is approximately 5 cycles/nm. (Image courtesy of A. Kirkland)

There is considerable scope for confusion of terminology in what follows. The diffractogram, which is produced by Fourier transforming the image of the amorphous carbon film, is a power spectrum in angular or spatial frequency. However, in the present context we wish to treat it as a slightly unusual image, (the circular or quasi-elliptical fringe pattern) which multiplicatively modulates a wide-band noise carrier. The image (diffractogram) coordinates will be polar $(r, \theta)$.

Figure 6-4 Diffractogram fit. The top right-hand half shows the original image; the bottom half shows a hand-fitted theoretical diffractogram. Note the grey spots in the original image. This is where the pre-processing algorithm has been applied to suppress spikes in the image due to crystalline inclusions in the carbon film. The image has also been aligned with the axes using the algorithm described in section 6.7.2. The original image is the one from which the calculations in sections 6.7.2, 6.7.3, and 6.7.5 were taken.

The signal to noise ratio in the diffraction pattern is quite poor. This is due in part to a relatively low number of electrons being detected. Although more electrons could be used, it is desirable to minimise the dose that the specimen receives. However, much of the poor signal to noise is due to the inhomogeneous nature of the scattering structures in the amorphous carbon film. Manual fits, such as that in Figure 6-4, have
been used previously, but computerised determination is also possible by using image processing techniques, such as those described in chapter 2.

6.4 Description of the Image Processing

The algorithm chosen for the analysis operates in three stages. The first is a preprocessing stage, which removes the central DC spike from the diffractogram and suppresses any other sharp peaks which are believed to be the result of crystalline inclusions in the carbon film, based on their grey level relative to their neighbours. The second and third stages then perform the fit to the data to determine the values for the various aberrations. The latter stages are based on correlation. The second stage uses the method presented in [97] and described in chapter 2, to find the axes of symmetry in the diffractogram. That is, the image is inverted along different radial axes, and then correlated with itself, according to:

\[
C(\phi) = \int_0^\pi \int_0^R (f_{x,y}(r,\phi + \phi_{22},+\theta) - f_{x,y}(r,\phi + \phi_{22},-\theta))^2 r\,dr\,d\theta
\]

where \( f_{x,y} \) is the diffractogram or the Fourier transformed image (\( \phi_{22} \) is described in chapter 2). This yields two correlation minima over a rotation of 180°, corresponding to the major and minor axis of the quasi-elliptical diffractogram. This angle is then used to rotate the image, using bilinear interpolation, so that its major and minor axes sit perpendicular to the \( x \) and \( y \) axes in the image.

The third stage then correlates the rotated image with a series of synthetically generated diffractograms, with different values of \( A_1 \) and \( C_1 \) (initially the manufacturer’s estimate of \( C_3 \) assumed to be correct). It then takes the minimum correlation score to be the best estimate of the values of \( A_1 \) and \( C_1 \).
### 6.5 Problem Level Analysis

#### 6.5.1 Parameters Affecting Performance

After viewing images and speaking with JEOL (UK) about lens aberrations, seven parameters that affect the accuracy of the IP system in determining the values of the aberration coefficients were identified. These are as follows:

<table>
<thead>
<tr>
<th>Image Parameter</th>
<th>Rank Sev.</th>
<th>Rank Occ.</th>
<th>Affect on Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fringe contrast, $\gamma$</td>
<td>=1</td>
<td>=1</td>
<td>Affects the contrast of the fringes on the FT image</td>
</tr>
<tr>
<td>Noise, $\sigma$</td>
<td>=1</td>
<td>=1</td>
<td>Creates speckled effect on diffractogram</td>
</tr>
<tr>
<td>Frequency Spectrum of the amorphous film, $f_i$</td>
<td>3</td>
<td>=1</td>
<td>Affects the contrast of the high frequency fringes</td>
</tr>
<tr>
<td>Values of the aberration coefficients, $A_1$, $C_1$, $C_3$.</td>
<td>=4</td>
<td>=1</td>
<td>Changes the shape of the diffractogram</td>
</tr>
<tr>
<td>Beam tilts, $\tau_x$, $\tau_y$</td>
<td>=4</td>
<td>6</td>
<td>Changes the shape of the diffractogram</td>
</tr>
<tr>
<td>Electron Wavelength, $\lambda$</td>
<td>=4</td>
<td>=1</td>
<td>Changes the scale of the diffractogram</td>
</tr>
<tr>
<td>Search range for aberration coefficients</td>
<td>7</td>
<td>7</td>
<td>No effect on image – Determines the size of the search space for correlation</td>
</tr>
</tbody>
</table>

Table 6-1 Parameters affecting the performance of the lens aberration determination system.

Each of the parameters has a significant affect on the image. The first three can be estimated quantitatively from the diffractogram. The beam tilt and electron wavelength are assumed to be already known. The effect of the other aberration coefficients cannot be calculated until they have been estimated by the system.

#### 6.5.2 Performance Metrics

The user performance metrics are the probability of the system finding the minima corresponding to the correct values of the aberration coefficients, and the errors in the estimates of the coefficients if the correct minimum is found. An estimate of the probability of finding the incorrect minimum, performed using the same analysis as in section 4.4.2, yields a signal to background response ratio of 7. This results in a reasonably low probability of finding the incorrect peak (though this depends upon the...
size of the search space). However here we are going to concentrate on the errors in the position of the minimum, assuming the correct minimum has been found.

### 6.5.3 Problem Classification

All the parameters are reasonably easy to measure or estimate and the problem is well constrained and amenable to analytical analysis. This problem falls into category two and an analytical model is developed to analyse the performance.

### 6.6 System Level Analysis

The system used here to determine the lens aberrations is fairly straightforward, and system modularisation is not essential for the performance analysis. However it is presented in this format here as a further illustration of the applicability of the modularisation technique to a variety of problems.

#### 6.6.1 Modularisation

The system breaks down readily into following modules, as described in the previous section.

- **Module 1. Spike Suppressor.** Removes the central DC spike from the diffractogram and suppresses any other sharp peaks which are believed to be the result of crystalline inclusions in the carbon film, based on their grey level relative to their neighbours.

- **Module 2. Inverter-Correlator.** (Measurement of astigmatic axis orientation \( \phi_{22} \)) Takes a single arbitrary image (i.e. a diffractogram). and folds the top half of the image onto the lower half and measures the cross-correlation of the two half-images. The same correlation measure \( C(\phi) \) is obtained for a range of new axes, each passing through the origin but rotated by some angle \( \phi \), for \( 0 < \phi < \pi \), according to:

\[
C(\phi) = \int_0^\pi \int_0^R \left( f_0(r, \phi + \phi_{22} + \theta) - f_0(r, \phi + \phi_{22} - \theta) \right)^2 r dr d\theta
\]

**eq(6-10)**
where $f_0(r, \theta)$ describes the original image data in terms of radial distance $r$ from the origin and polar angle $\theta$, while $\phi$ is the angle through which the image is rotated before folding about the $x$-axis and correlating.

Module 3. Minima Finder. For an image obtained from tilted-beam recording there should be two principal minima in $C(\phi)$, with almost equal values and separated by 90 degrees. These represent the folding of the pattern about its long and short axes. This module finds the two minima. However which minimum is which (the long and short axes) cannot be determined at this point. The value of $\phi_{22}$, with an uncertainty of 90°, is then passed to the next module.

Module 4. Image Rotator. This rotates the image, using bilinear interpolation, through the angle $\phi_{22}$, so that the image is now orientated with the major and minor axes parallel to the $x$ and $y$ axes in the image.

Module 5. Diffractogram Generator and Correlator. This generates a series of synthetic diffractograms, with different values of $A_1$ and $C_1$ ($C_3$ is assumed to be known) and correlates them with the orientated image. Negative values of $A_1$ are also used, as this allows the ambiguity between the major and minor axes in the diffractogram to be resolved.

Module 6. Minimum Finder. This finds the minimum in the correlation surface $C(A_1,C_1)$ generated by module 4, to produce the estimates of the values of $A_1$ and $C_1$.

A block diagram of the information stream is shown in Figure 6-5.

![Figure 6-5 Block diagram of the lens aberration determination system](image-url)
6.6.2 Interaction and Propagation of Parameters

The external and performance parameters interact with the different modules as follows.

Module 1. The cleaning and removing of the spikes is affected by the size and number of the spikes. It is assumed here that a certain amount of user interaction may be required to determine the correct number of spikes to be removed.

Module 2. The generation of the correlation surface is affected by the following parameters: Fringe contrast, \( \gamma \); Noise, \( \sigma \); Frequency Spectrum of the amorphous film, \( f_I \); Values of the aberration coefficients, \( A_1, C_1, C_3 \); Beam tilts, \( \tau_x, \tau_y \); Electron Wavelength, \( \lambda \).

Module 3. The accuracy of the minima finder is affected by the noise in the correlation surface \( C(\phi) \), the shape of the correlation minima and the depth of the minima relative to the spurious minima, and the correlation signal depth.

Module 4. The accuracy of the orientation of the image is dependent upon the accuracy of the estimate of \( \phi_{22} \).

Module 5. This is dependent on all the same parameters as Module 1: Fringe contrast, \( \gamma \); Noise, \( \sigma \); Frequency Spectrum of the amorphous film, \( f_I \); Values of the aberration coefficients, \( A_1, C_1, C_3 \); Beam tilts, \( \tau_x, \tau_y \); Electron Wavelength, \( \lambda \). It is also dependent on the accuracy of the image orientation performed by module 3.

Module 6. The accuracy of the minimum found is dependent upon noise in the correlation surface, \( C(A_1, C_1) \), the shape of the correlation minima and the correlation signal depth.
6.7 Performance Characterisation of System

It is now possible to analyse the performance of each of the modules in turn at the algorithm level, to determine the transfer function between the input parameter(s), the input quality measure(s) and the output quality measure(s).

6.7.1 Module 1. Preprocessing – Spike Supresser

As illustrated in Figure 6-3, it is necessary to remove the high-intensity central spot, and replace it by the mean of the neighbouring points. This reduces bias effects when measuring other image parameters. If there are any crystalline inclusions in the image of the amorphous carbon film (as seen for example in Figure 6-3) then these will produce diffraction peaks away from the centre of the image. It is straightforward to remove a pre-specified small number of such high-intensity spots, but quite difficult to automate the estimation of whether such spots exist or not. We are not going to analyse the spot remover here, and assume that if used, some user interaction may be required. For the remainder of this analysis we assume that such spots have been successfully removed.

6.7.2 Module 2: Inverter and Correlator

The Correlator Module is as follows:

![Diagram of Correlation Map and Input Parameters]

**Figure 6-6** Parameter interaction with the correlator module. The top arrows show the information stream through the module. The remaining arrows indicate the external parameters acting on this module and the quality propagation parameters which we need to estimate the performance of the following module.
The transfer function between $s(x,y)$ and $c(\phi)$ is given in eq(2-3). We need to evaluate the transfer function between the external parameters and the quality propagation parameters. In addition, the analysis which follows shows how it is possible to obtain an approximate estimate of the two-fold astigmatism coefficient $A_1$, which can be used subsequently to reduce the search space when trying to find a more precise value for $A_1$.

Figure 6-7 illustrates how the correlation varies with $\phi$. The angular location of the two minima, which are 90 degrees apart, represent the two possible fold angles for which the quasi-elliptical fringe pattern overlaps most with itself. A second trace shows the effect of smoothing the diffractogram by convolving it with a Gaussian mask ($\sigma = 3$ pixels) before performing the correlation. In practice, smoothing caused the minima to drift due to variations in the mean value of $C(\phi)$, away from the minima themselves. This introduced further errors into the system, despite appearing to smooth the minima.

Figure 6-7 Correlation score $C(\phi)$ versus rotation angle $\phi$ of diffractogram before folding. The upper trace is the result of correlating the original (but interpolated) data; the spikes at readings $\phi = 0^0$ and $90^0$ occur because no interpolation is needed for these two rotation angles. The lower trace shows the correlation score when the data were smoothed before correlation by convolution with a gaussian mask with a standard deviation of approximately 3 pixels. Both sets of data are unnormalised.
6.7.2.1 Explanation of Notation

As already indicated in equations 6-8 and 6-10, polar co-ordinates \((r, \theta)\) will be used to define points in the diffractogram image space, even though the diffractogram is a representation of the power at different spatial or angular frequencies in the original amorphous carbon film. The diffractogram may be considered to be an image in its own right, with the fringe ‘signal’ modulating a ‘noisy’ carrier. By ‘signal’ we mean either the \(\sin^2\) envelope or one of its defining parameters. By ‘noise’ we mean the power spectrum of the specimen, which would have been obtained if there had been no aberrations in the imaging system. In order to proceed further, it is necessary to develop a model of the diffractogram structure.

6.7.2.2 Model of Diffractogram

We model the normalised diffractogram image, \(f_6(r, \theta)\), by the product of three factors:

\[
f_6(r, \theta) = f_1(r, \theta) f_2(r, \theta) f_3(r, \theta)
\]

where

- \(f_1\) describes a smoothly varying, approximately circular envelope,
- \(f_2\) describes the sum of a constant-amplitude fringe pattern and a constant ‘pedestal’, and
- \(f_3\) is specified by the sum of a constant component (unity) and a stochastic component with standard deviation \(\sigma_3\) and correlation length \(d_3\).

The three factors are illustrated in Figure 6-8.
The three-component model consists of a smooth envelope function $f_1$ (top left), an adjustable-contrast fringe pattern $f_2$ (top right) and a noise component $f_3$ (bottom left). The product of these three factors gives the synthetic diffractogram pattern $f_0$ (bottom right).

The envelope $f_1$ could be described by a function which decays smoothly and monotonically with radius. The envelope is a result of the slight spatial and temporal incoherence in the wave, due to the finite size of the illumination aperture and an energy width, $\Delta E$, in the electron emission process of 0.2-0.3eV for a field emission electron gun. It can be shown, for example in [94], that this results in an approximately gaussian envelope function with equal standard deviation $\sigma_E$ along the $x$ and $y$ axes. In practice, it is found that a circular ‘top hat’ function provides an adequate model for $f_1$. The modulation function, $f_2$ describes the CTF:
\[ f_2(r, \theta) = (1 - \gamma) + 2\gamma \sin^2(ar^2 + br^2 \cos(2(\theta - \phi_{22})) + cr^4) \]
\[ = 1 + \gamma \cos(2(ar^2 + br^2 \cos(2(\theta - \phi_{22})) + cr^4) \]

eq(6-12)

where \( \gamma \) is a measure of the fringe pattern modulation depth. The new variables \( a, b \) and \( c \), introduced here for notational simplicity, are proportional to the defocus, primary astigmatism and spherical aberration respectively:

\[ a = -\pi \lambda C_1 / 2; \quad b = -\pi \lambda A_1 / 2; \quad c = \pi \lambda^3 C_3 / 2 \]

eq(6-13)

The \( f_3 \) factor describes the object power spectrum that would be obtained with no TEM aberrations and with infinitely high image resolution. This may be written in the form:

\[ f_3(r, \theta) = 1 + \xi_3 \]

eq(6-14)

where \( \xi_3 \) represents a quasi-stochastic process with zero mean, standard deviation \( \sigma_3 \) and correlation length \( d_3 \). In more well illuminated images, \( \xi_3 \) has an approximately Gaussian distribution. However with low illumination, \( \xi_3 \) is closer to a Poisson distribution. Such images are not considered here. The value of \( d_3 \) in the original diffractogram is nearly zero, but it can be shown that digital interpolation, which is needed to produce rotated versions of the pattern, produces finite values for \( d_3 \) and, more importantly, reduces the standard deviation. The value of the variance \( \sigma_3^2 \) will depend upon the assumed form for the theoretical ‘noise’ probability distribution. To proceed further, we assume that the envelope function \( f_i \) can be approximated by a constant, \( \bar{f}_i \), over a circle of maximum radius \( R \), and zero outside. \( R \) can be estimated experimentally by integrating the encircled diffractogram until the integral reaches, for example, 90% of the maximum. The integral of \( f_0 \) over the circle of radius \( R \) pixels is then given approximately by \( \pi R^2 \bar{f}_i \) and the integral of \( f_0^2 \) by

\[ \int_0^R \int_0^{2\pi} f_0^2 r \, dr \, d\theta = \pi R^2 \bar{f}_i^2 (1 + \sigma_3^2)(1 + \gamma^2 / 2) \]

eq(6-15)
The variance $\sigma_3^2$ can be measured automatically by scanning a small patch (e.g. 8 by 8 pixels) over the diffractogram, excluding the central maximum. For the off-axis diffractograms in Figure 6-3, $\sigma_3^2$ was found to vary from 1.2 – 1.3, except very near the edge of the diffractogram, where it rose to about 1.7. The fringe amplitude $\gamma$ can be measured by smoothing the image with a gaussian kernel with $\sigma = 3$ pixels to reduce the 'noise' component $\xi_3$. The first maximum and minimum (over 4 by 4 pixel windows) are then found along two orthogonal directions and the ratio of the average maximum to average minimum values is $(1+\gamma)/(1-\gamma)$. For the central fringes, values of $\gamma$ in the range $0.65 \pm 0.06$ were found for the images in Figure 6-3. Hence from eq(6-15), the integral of $f_0^2$ is approximately $2.8\pi R^2 f_1$.

The correlation of the top and bottom halves of the rotated image can be expressed in terms of $f_1$, $f_2$ and $f_3$:

$$C(\phi) = \int_0^\pi \int_0^R \left( f_0(r,\phi+\phi_22,+\theta) - f_0(r,\phi+\phi_22,-\theta) \right)^2 r dr d\theta$$

$$= \int_0^\pi \int_0^R \left( f_{2\text{top}} f_{3\text{top}} - f_{1\text{bottom}} f_{2\text{bottom}} f_{3\text{bottom}} \right)^2 r dr d\theta$$

$$= \frac{f_1^2}{\pi} \int_0^\pi \int_0^R \left( f_{2\text{top}} f_{3\text{top}} - f_{2\text{bottom}} f_{3\text{bottom}} \right)^2 r dr d\theta$$

eq(6-16)

where the 'top' and 'bottom' suffixes refer to the upper and lower half-planes of the rotated image. If the image had no noise component $\xi_3$ then the value of $C(\phi)$ would be zero (a perfect match) for $\phi = -\phi_22$ and $\phi = \pi - \phi_22$. The presence of the noise component raises the value of $C(\phi)$ from this minimum value.

We now expand the integrand in eq(6-16), using the shorthand symbols $\delta$ and $\epsilon$ to represent rotated versions of the arguments, over the top and bottom halves of the field, of the cosine function in expression eq(6-12):
\[ \delta = 2(ar^2 + br^2 \cos(2(\phi + \phi_{33} + \theta)) + cr^4) \quad \text{(top half of field)} \]
\[ \varepsilon = 2(ar^2 + br^2 \cos(2(\phi + \phi_{33} - \theta)) + cr^4) \quad \text{(bottom half of field)} \]

\text{eq}(6-17), \quad \text{eq}(6-18)

The bracketed term in eq(6-16) may be written in the form

\[ (\ldots) = (1 + \gamma \cos(\delta))(1 + \xi_{3\text{top}}) - (1 + \gamma \cos(\varepsilon))(1 + \xi_{3\text{bottom}}) \]

\text{eq}(6-19)

using expressions eq(6-12) and eq(6-14) for \( f_2 \) and \( f_3 \). Squaring this expression gives:

\[ (\ldots)^2 = \gamma^2 (\cos \delta - \cos \varepsilon)^2 + (\xi_{3\text{top}} - \xi_{3\text{bottom}})^2 + \gamma^2 (\xi_{3\text{top}} \cos \delta - \xi_{3\text{bottom}} \cos \varepsilon)^2 \]
\[ + 2\gamma (\cos \delta - \cos \varepsilon)(\xi_{3\text{top}} - \xi_{3\text{bottom}}) \]
\[ + 2\gamma (\xi_{3\text{top}} \cos \delta - \xi_{3\text{bottom}} \cos \varepsilon)(\xi_{3\text{top}} - \xi_{3\text{bottom}}) \]
\[ + 2\gamma^2 (\xi_{3\text{top}} \cos \delta - \xi_{3\text{bottom}} \cos \varepsilon)(\cos \delta - \cos \varepsilon) \]

\text{eq}(6-20)

To find the expectation value of \( C(\phi) \), we note that \( \xi_{3\text{top}} \) and \( \xi_{3\text{bottom}} \) are independent zero-mean processes (except for a vanishingly small number of points near the fold axis) so that their integrals and cross-product integrals average to zero. One therefore obtains:

\[ \overline{C(\phi)} = (\pi R^2 f_i^2 / 2) (\gamma^2 (\cos \delta - \cos \varepsilon)^2 + 2\sigma_3^2 + 2\gamma^2 (\cos \delta \sigma_3^2) \sigma_3^2) \]

\text{eq}(6-21)

where the primes on the variances indicate that they may differ (due to interpolation) from the variance \( \sigma_3^2 \) of the original pattern, but are otherwise identical in the top and bottom parts of the pattern. Away from minima or maxima, that is, when \( \delta \) and \( \varepsilon \) are significantly different, \( C(\phi) \) reduces to:

\[ \overline{C(\phi)} = (\pi R^2 f_i^2 / 2)(\gamma^2 + 2\sigma_3^2 + \gamma^2 \sigma_3^2) \]

\text{eq}(6-22)
and at the minima, where $\delta$ and $\epsilon$ are identical, it becomes:

$$C(\phi) = \pi R^2 f_1^2 \sigma_3^2 (1+\gamma^2 / 2)$$

eq(6-23)

where $\sigma_3^2$ is the (identical) variance of the upper and lower halves of the rotated image. In general $\sigma_3$ will differ in value from the $\sigma_3$ of the original (unrotated) image, because the data are digitised on a square grid and interpolation is needed to generate rotated versions. The $\xi_3$ component of the initial data can be modelled as white noise, and is therefore uncorrelated from point to point, and $\delta_3 = 0$. However if bilinear interpolation is used then one may show that the resulting data have an average inter-pixel correlation $\rho = 0.25$ and a reduced variance $\sigma_{\text{interpol.}}^2 = (4/9)\sigma_{\text{original}}^2$. Hence

$$\sigma_{\text{interpol.}}^2 = 0.444\sigma_{\text{original}}^2$$

except for the two angles $\phi = 0^\circ$ or $90^\circ$. Three different expectation values for $C(\phi)$ therefore exist:

1. at the points where $\phi = 0^\circ$ or $90^\circ$:
   $$\bar{C}(\phi) = (\pi R^2 f_1^2 / 2)(\gamma^2 + (2 + \gamma^2)\sigma_3^2)$$

2. the mean reading, excluding $0^\circ$ or $90^\circ$:
   $$\bar{C}(\phi) = (\pi R^2 f_1^2 / 2)(\gamma^2 + (2 + \gamma^2)\sigma_3^2)$$

3. at the correlation minima:
   $$\bar{C}(\phi) = (\pi R^2 f_1^2)(1+\gamma^2 / 2)\sigma_3^2$$

Figure 6-9 and Figure 6-10 plot $C_{\text{peak}} / C_{\text{mean}}$ and $C_{\text{mean}} / C_{\text{minima}}$ as a function of the amplitude $\gamma$. 

140
Figure 6-9 Predicted ratio of correlation value at peak \( (C_{\text{peak}}) \) to mean correlation value away from peak or minima \( (C_{\text{mean}}) \), as a function of the fringe amplitude \( \gamma \), for several values of the noise variance, \( \sigma_3^2 \).

Figure 6-10 Predicted ratio of the mean correlation value away from peaks or minima \( (C_{\text{mean}}) \) to the expected correlation value at the minima \( (C_{\text{min}}) \), as a function of the fringe amplitude \( \gamma \), for several values of the variance \( \sigma_3^2 \).
From Figure 6-7, we see that the experimental (pre-normalised) value for $C_{\text{peak}} / C_{\text{mean}}$ is $5\,600\,000/2\,800\,000 = 2.0$ and the measured variance of the diffractogram was approximately 1.25. Using these two parameter values in Figure 6-9 then predicts that the value of the fringe amplitude $\gamma$ should be about 0.63, which agrees quite well with the experimental value of $\gamma = 0.65 \pm 10\%$ for the central region of Figure 6-4. Similarly, the experimental value for $C_{\text{mean}} / C_{\text{minima}}$ is $2\,800\,000/2\,250\,000 = 1.24$. Assuming the same values of variance (1.2 - 1.3) and fringe amplitude ($\gamma = 0.65 \pm 10\%$), then Figure 6-10 predicts that $C_{\text{mean}} / C_{\text{minima}}$ should be $1.28 \pm 0.08$ The agreement is quite good, given the simplifying assumptions used to estimate $C_{\text{minima}}$.

6.7.2.3 Approximate Estimates for $A_1$

An initial, approximate estimate of the astigmatism coefficient $A_1$ can be made by measuring the width of the minimum of the correlation curve. We show in this section that the width of the correlation minimum is inversely proportional to the first-order astigmatism coefficient $A_1$. This approximate estimate for $A_1$ may then be used to constrain the range of values which subsequently need to be examined in template matching.

Consider again the expansion in eq(6-20) for the bracketed term in the correlation integral:

$$(\ldots)^2 = \gamma^2(\cos \delta - \cos \varepsilon)^2 + (\xi_{\text{top}} - \xi_{\text{bottom}})^2 + \gamma^2(\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \varepsilon)^2$$

$$+ 2\gamma(\cos \delta - \cos \varepsilon)(\xi_{\text{top}} - \xi_{\text{bottom}})$$

$$+ 2\gamma(\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \varepsilon)(\xi_{\text{top}} - \xi_{\text{bottom}})$$

$$+ 2\gamma^2(\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \varepsilon)(\cos \delta - \cos \varepsilon)$$

**eq(6-27)**

and assume for the moment that the noise terms are all zero. Eq(6-27) then reduces to:

$$(\ldots)^2 = \gamma^2(\cos \delta - \cos \varepsilon)^2 = \gamma^2(\cos^2 \delta + \cos^2 \varepsilon) - 2\gamma^2 \cos \delta \cos \varepsilon$$

**eq(6-28)**
The integral of the first bracketed term is a constant, equal to \( \pi R^2 \gamma^2 f_0^2 / 2 \), but the integral of the cross-product term depends on the rotation angle. We expand the second term in eq(6-28), using the relation:

\[
2\cos \delta \cos \epsilon = \cos(\delta - \epsilon) + \cos(\delta + \epsilon)
\]

**eq(6-29)**

where

\[
\delta + \epsilon = 4ar^2 + 2br^2 \cos(2(\phi + \phi_{22} + \theta)) + 2br^2 \cos(2(\phi + \phi_{22} - \theta)) + 4cr^4
\]

**eq(6-30)**

and

\[
\delta - \epsilon = 2br^2 (\cos(2(\phi + \phi_{22} + \theta)) - \cos(2(\phi + \phi_{22} - \theta)))
\]

**eq(6-31)**

In eq(6-31) we then use the standard formula for the difference of two cosines. If we assume that we are in the region of the axis of symmetry of the pattern, then we may write \( \phi + \phi_{22} = \Delta \phi \), where \( \Delta \phi \) is the small angle between the true axis of symmetry and the axis currently being used to fold the image. Eq(6-31) then becomes:

\[
\delta - \epsilon = -2br^2 (2 \sin(2\theta) \sin(2\Delta \phi))
\]

\[
= -8br^2 \Delta \phi \cos(2\theta)
\]

**eq(6-32)**

(since \( \sin(2\Delta \phi) \approx 2\Delta \phi \)), while eq(6-30) becomes:

\[
\delta + \epsilon = 4ar^2 + 4cr^4 + 4br^2 \cos(2(\Delta \phi + 2\theta)) + \cos(2(\Delta \phi - 2\theta)) + 4br^2 \cos(2(\theta + 2\Delta \phi))
\]

\[
= 4ar^2 + 4cr^4 + 4br^2 \cos(2\Delta \phi) \cos(2\theta)
\]

\[
\approx 4ar^2 + 4cr^4 + 4br^2 \cos(2\theta)
\]

**eq(6-33)**

since \( \cos(\Delta \phi) \approx 1 \) for small \( \Delta \phi \). Hence the integral of the second term in eq(6-28) may be written:

\[
-\gamma^2 \iint 2\cos \delta \cos \epsilon r dr d\theta \\
-\gamma^2 \iint \cos(4ar^2 + 4cr^4 + 4br^2 \cos(2\theta)) r dr d\theta - \gamma^2 \iint \cos(-8br^2 \Delta \phi \sin(2\theta)) r dr d\theta
\]

**eq(6-34)**
by a suitable choice of origin for $\theta$. The correlation integral $C(\Delta \phi)$, for small angular deviations $\Delta \phi$ from the correlation minimum, is therefore given by:

$$ C(\phi) = (\pi R^2 \gamma^2 f_1^2 / 2) - \gamma^2 f_1^2 \int \cos(4ar^2 + 4cr^4 + 4br^2 \cos(2\theta))rdrd\theta $$

$$ -\gamma^2 f_1^2 \int \cos(-8br^2 \Delta \phi \sin(2\theta))rdr\theta $$

eq(6-35)

The second term integrates approximately to a constant. The third term determines the shape of the correlation minima. Consider the third integral term in eq(6-35). Setting $\rho = r^2$ and $\psi = 2\theta$, gives:

$$ I_2(\Delta \phi) = -\left(\gamma^2 f_1^2 R^2 / 4\right) \int_0^{2\pi} \int_0^R \cos(8b \Delta \phi \rho \sin(\psi)) d\rho d\psi $$

eq(6-36)

where $R$ is the radius at which the integral of $f_0$ reaches (eg) 90% of the maximum value. Integrating over $\rho$ yields

$$ I_2(\Delta \phi) = -\left(\gamma^2 f_1^2 R^2 / 4\right) \int_0^{2\pi} \sin(GR^2 \sin(\psi)) d\psi $$

$$ \frac{GR^2 \sin(\psi)}{GR^2 \sin(\psi)} $$

eq(6-37)

where $G = 8b \Delta \phi$.

The integrand is therefore a sinc function. Since $|\sin\psi|$ is never greater than unity, then for values of $G$ such that $GR^2 << \pi$, the sinc function can be approximated by the first two terms in its series expansion, so that:

$$ I_2(\Delta \phi) = -\left(\gamma^2 f_1^2 R^2 / 4\right) \int_0^{2\pi} \frac{G^2 R^4 \sin^2 \psi}{3!} d\psi = -\left(\pi \gamma^2 f_1^2 R^2 / 2\right) \left(1 - G^2 R^4 / 12\right) $$

eq(6-38)

However:
\[ G = 8b\Delta \phi \quad ; \quad b = -\pi \lambda A_i \]

\[ \therefore G^2 = 16\pi^2 \lambda^2 A_i^2 (\Delta \phi)^2 \]

and hence

\[ I_2(\Delta \phi)^2 = (\pi \gamma^2 f_1^2 R^2 / 2) \left(1 - \left(4\pi^2 R^4 \lambda^2 A_i^2 / 3\right)(\Delta \phi)^2\right) \]

\[ \text{eq}(6-39) \]

This has the quadratic variation with \( \Delta \phi \) to be expected in general for a correlation maximum or minimum. Also, the astigmatism coefficient \( A_j \) is inversely proportional to the width of the correlation minimum.

If we now consider the data after it has been normalised to unit variance (effectively dividing the integral by \( \pi R^2 f_0^2 \)), we get from eq(6-39)

\[ I_2(\Delta \phi)^2 = \frac{1}{\pi R^2 f_0^2} \left(\pi \gamma^2 f_1^2 R^2 / 2\right) \left(1 - \left(4\pi^2 R^4 \lambda^2 A_i^2 / 3\right)(\Delta \phi)^2\right) \]

\[ \text{eq}(6-40) \]

which evaluates to

\[ I_2(\Delta \phi)^2 = \frac{\gamma^2 \left(3 - \left(4\pi^2 R^4 \lambda^2 A_i^2\right)(\Delta \phi)^2\right)}{3(1 + \sigma^2)(2 + \gamma^2)} \]

\[ \text{eq}(6-41) \]

We now use least squares to fit a quadratic function to the bottom of the correlation minimum, of the form

\[ C_0(\Delta \phi) = \alpha(\Delta \phi)^2 + \beta(\Delta \phi) + \chi \]

\[ \text{eq}(6-42) \]

Then by comparing the quadratic term, \( \alpha \), in eq(6-42) with eq(6-41) gives:

\[ \alpha = \frac{4\pi^2 R^4 \gamma^2 \lambda^2 A_i^2}{3(1 + \sigma^2)(2 + \gamma^2)} \]

\[ \text{eq}(6-43) \]

The (apparent) astigmatism coefficient, \( A_j \), can therefore be estimated from:
Figure 6-11 By fitting a quadratic curve to the minimum of the correlation function, an approximate estimate of the apparent primary astigmatism coefficient $A_1$ can be obtained using eq(6-44). This provides a useful constraint during the subsequent template matching process to obtain a more accurate estimate of $A_1$.

Figure 6-11 shows the results of fitting a quadratic curve to determine $\alpha$ and hence $A_1$ for an example diffractogram. The $\xi_3$ noise term has no first-order effect on the value of $\alpha$.

This yields an estimate of $\alpha = 5.3$, giving a value for $A_1$ from eq(6-44) of 46nm. This compares moderately well with the value for $A_1$ of 30nm determined using conventional manual fitting to the diffractogram. (Note that this is not a final calculation of $A_1$. It is intended as a rough initial estimate, to enable the search space used in the subsequent analysis to be refined. Hence a seemingly quite large discrepancy at this point is acceptable.)

We note also that the mean value of the integral, away from the minima, is estimated to be
\[
\frac{f_1^2}{f_0^2} = \frac{1}{(1 + \sigma^2)(1 + \gamma^2 / 2)} = 0.37.
\]

eq(6-45)

From Figure 6-11, the mean value is 0.35.

6.7.3 Module 3: Minima Finder

The second module then finds the minima in the correlation surface, \(C_0(\Delta \phi)\). The input quality measures are the noise in the correlation surface, \(\sigma_c\), and the shape of the correlation minima, \(\alpha\). The output metric is the standard deviation on the measure of \(\phi_{22}, \sigma(\phi_{22})\).

6.7.3.1 Effect of Noise on the Accuracy in Measuring \(\phi_{22}\)

The calculation of \(A_1\) given above ignored the effects of the noise terms. These have little effect on the measurement of the width, but they affect the angular accuracy with which the correlation minimum, and hence \(\phi_{22}\), can be estimated. The magnitude of the correlation noise in the vicinity of the correlation minimum can be estimated from eq(6-20) for the integrand in the correlation integral, reprinted below:

\[
(....)^2 = \gamma^2 (\cos \delta - \cos \epsilon)^2 + (\xi_{\text{top}} - \xi_{\text{bottom}})^2 + \gamma^2 (\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \epsilon)^2 \\
+ 2\gamma (\cos \delta - \cos \epsilon)(\xi_{\text{top}} - \xi_{\text{bottom}}) \\
+ 2\gamma (\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \epsilon)(\xi_{\text{top}} - \xi_{\text{bottom}}) \\
+ 2\gamma^2 (\xi_{\text{top}} \cos \delta - \xi_{\text{bottom}} \cos \epsilon)(\cos \delta - \cos \epsilon)
\]

At the correlation minimum \(\delta = \epsilon\), and eq(6-20) reduces to:

\[
(....)^2 = (1 + \gamma^2 \cos^2 \delta)(\xi_{\text{top}} - \xi_{\text{bottom}})^2
\]

eq(6-46)

Although this has been expressed as a continuous function, it is evaluated as a discrete summation over the image. If the image contains \(N\) pixels, where \(N = \sqrt{\pi R^2 / d}\), and \(d\) is the size that one pixel represents in frequency space (\(= 15 \times \))
\(10^9/256 = 5.86 \times 10^7 \text{ cycles/m}, \) then at the minimum, the expectation value of \(C, \langle C \rangle, \) is:
\[
\langle C \rangle = 2N f_1^2 f_2^2 \sigma_3^2
\]
\[\text{eq}(6-47)\]
and
\[
\langle C^2 \rangle = f_1^2 f_2^2 (4N^2 \sigma_3^4 + 4N \sigma_3^4)
\]
\[\text{eq}(6-48)\]
Hence
\[
\text{var}(C) = \langle C^2 \rangle - \langle C \rangle^2 = 4f_1^2 f_2^2 N \sigma_3^4
\]
\[\text{eq}(6-49)\]
and
\[
\sqrt{\text{var}(C)} = \sigma_c = 2\sigma_3^2 f_1 f_2 \sqrt{N}
\]
\[\text{eq}(6-50)\]
Scaling this to match the actual data, which have been normalised to \(\sum_{i=1}^{N} f_0^2 = 1,\) and noting from eq(6-45) that \(\overline{f}_0^2 = \overline{f}_1^2 (1 + \sigma_3^2)(1 + \gamma^2 / 2),\) and from eq(6-12), that \(f_2^2 = (1 + \gamma^2 / 2),\) we obtain
\[
\sigma_c = \frac{2\sigma_3^2}{\sqrt{N}(1 + \sigma_3^2)}
\]
\[\text{eq}(6-51)\]
For determining the value of \(\phi_{22},\) a relatively small value of \(R (R = 4.7 \text{cycles/nm})\) was chosen to avoid skewing effects due to a slight asymmetry in the outer regions of the envelope. This yielded an estimated value for \(\sigma_c\) of 0.008, using a value of 1.25 for \(\sigma_3^2.\)
The measured value of \(\sigma_c\) from the real data in Figure 6-11 is approximately 0.006.
Figure 6-12 Illustration of the effect of noise in uncertainty in the position of the correlation minima. The diagram on the left shows the noise in the correlogram, which results in an uncertainty in $C(\phi)$ (in the vertical direction). This in turn causes an uncertainty in the position of the minima in the horizontal direction.

Now that $\sigma_c$ has been evaluated, it is possible to estimate its effect on the location of the correlation minimum. Figure 6-12 illustrates how the probability of finding the minimum at some angular location $\Delta\phi$ away from the ‘true’ minimum is affected by the shape $C_0(\Delta\phi)$ of the noise-free correlation function and by the standard deviation $\sigma_c$.

Relative to the expectation value of the correlation minimum, $C_0(\Delta\phi)$ is a simple quadratic function:

$$C_0(\Delta\phi) = \alpha(\Delta\phi)^2 + \beta\Delta\phi + \chi$$

**eq(6-52)**
where $\Delta \phi$ is the angular displacement from the minimum, while $\alpha$ has been determined from Figure 6-11 and $\beta$ and $\chi$ go to zero by a suitable choice of origin. In the presence of noise with standard deviation $\sigma_c$, given by eq(6-51), this $\delta$-function distribution is spread out along the $C$ axis. The probability of observing a particular value of $C$, for a given value of $\Delta \phi$, is given by:

$$p(C)dC = \frac{1}{\sqrt{2\pi} \sigma_c} \exp\left(-\frac{(C - C_0(\Delta \phi))^2}{2\sigma_c^2}\right)$$

eq(6-53)

if a Gaussian noise distribution is assumed.

The probability of observing a correlation value, at some angle $\Delta \phi$, which is lower than the noise-free correlation minimum is given by:

$$p(C(\Delta \phi < 0) = \int_{-\infty}^{0} p(C)dC = \int_{C_0}^{\infty} n(0, \sigma_c)dC = (1/2)\text{erfc}(C_0(\Delta \phi)\sqrt{2} / \sigma_c)$$

eq(6-54)

and hence the probability $p(\Delta \phi)$ of finding the minimum value of the correlation function at an angle $\Delta \phi$ away from the noise-free minimum is:

$$p(\theta) = \text{erfc}(\Delta \phi^2 \sqrt{2} \alpha / \sigma_c) \int_{\Delta \phi}^{\infty} \text{erfc}(\Delta \phi^2 \sqrt{2} \alpha / \sigma_c)d\theta$$

eq(6-55)

The denominator evaluates to $2(\pi \sqrt{2} \alpha / \sigma_c)^{-1/2} \gamma(3/4, \infty) = 1.1627(\sigma_c / \alpha)^{1/2}$, where $\gamma$ is the alternative incomplete Gamma function [98].

The standard deviation of the probable error in estimating the angle, $\phi_{22}$, $\sigma(\phi_{22})$, is thus approximately:

$$\sigma(\phi_{22}) = 0.378(\sigma_c / \alpha)^{1/2}$$

eq(6-56)
where \( \alpha \) is given by fitting eq(6-52) to the graph of \( C(\Delta \phi) \), and \( \sigma_c \) is given by eq(6-51). For the diffractogram used to generate Figure 6-11, the estimated value of \( \sigma(\phi_{22}) \) is 14 mrad.

6.7.4 Module 4: Image Rotator

The image rotator is assumed to not introduce any additional errors, i.e. the error in the rotation angle is the same as the error in \( \phi_{22} \), that is \( \sigma(\phi_{22}) \). Thus its effect on \( \phi_{22} \) can be neglected. Only the effect on \( \sigma_3 \) due to the bilinear interpolation analysed in section 6.7.2.2 need be considered.

6.7.5 Modules 5 and 6: Diffractogram Generator, Correlator and Minimum Finder.

The operation of the Correlator and Minimum Finder are so closely linked that they are considered here as one module.

![Diagram](image)

**Figure 6-13** The parameter interaction with the correlator and minimum module. The relevant parameters are the same as before, the only difference being the output is a two-dimensional correlation surface in \( A_1 \) and \( C_1 \).

A range of possible values for \( A_1 \) and \( C_1 \) is chosen, assuming that the value of \( C_3 \) is known and that the rotation due to astigmatism has been removed. A set of synthetic template patterns which span the expected range of \( A_1 \) and \( C_1 \) is produced and correlated against the diffractogram. In section 6.5.2 it was noted that the signal to
clutter ratio was approximately 7. The size of the search space for the correlation minimum is not known, however it is assumed here that the resulting two-dimensional correlation map will have a unique minimum within the expected range. The probability of the incorrect peak being found will not be considered. The $A_1$ range can be estimated from the $C(\phi)$ versus $\phi$ curve, as shown in section 6.7.2.3, but the $C_1$ range is more problematic. In principle it is necessary to span a large range of positive and negative defocus values.

The analysis of modules 5 and 6 follows similar lines to the analysis of modules 2 and 3. The main difference is that instead of correlating two halves of the diffractogram, we create a set of synthetic, noise-free diffractograms, using different values for the aberrations, and correlate each of them in turn with the diffractogram being analysed. Again the square of differences correlation measure is used. The minimum in the correlation surface is then taken to correspond to the true values of the astigmatism and defocus.

The same model of the diffractogram is used:

$$f_0(r, \theta) = f_1(r, \theta) f_2(r, \theta) f_3(r, \theta)$$

where

$$f_2(r, \theta) = (1 - \gamma) + 2\gamma \sin^2(ar^2 + br^2 \cos(2\theta) + cr^4)$$

$$= 1 + \gamma \cos(2(ar^2 + br^2 \cos(2\theta) + cr^4))$$

and

$$f_3(r, \theta) = 1 + \xi,$$

The expression for the square-of-differences correlation function is therefore:
\[ C(a', b', c') = \int_0^{2\pi} \int_0^R \left( f_0(r, \theta, a, b, c) - f_0(r, \theta, a', b', c') \right)^2 r dr d\theta \]

\[ = \int_0^{2\pi} \int_0^R \left( f_1 f_2 f_3 - f'_1 f'_2 \right)^2 r dr d\theta \]

\[ \text{eq(6-57)} \]

where \( f_1 \) is the template envelope function (which is estimated during the process for finding \( \phi_{22} \)) and \( f_2 \) is the current estimated fringe modulation pattern used to form the template:

\[ f_2' (r, \theta) = (1 - \gamma) + 2\gamma \sin^2 (a' r^2 + b' r^2 \cos(2\theta) + c' r^4) \]

\[ = 1 + \gamma' \cos(2(a' r^2 + b' r^2 \cos(2\theta) + c' r^4)) \]

\[ \text{eq(6-58)} \]

The factors \( a' \), \( b' \) and \( c' \) are as defined in eq(6-13), but the primes indicate that the template image is being referred to. \( \gamma' \) is the estimate of the fringe modulation (note that there is no \( f_3 \) noise term in the template). If \( f_1 = f_1' \) then \( C(a', b', c') \) becomes (pre-normalisation):

\[ C(a', b', c') = \int_0^{2\pi} \int_0^R \left( f_1 f_2 f_3 - f'_1 f'_2 \right)^2 r dr d\theta \]

\[ \text{eq(6-59)} \]

We now expand the integrand, using the shorthand symbols \( \delta \) and \( \varepsilon \), in a slightly different way from before, to represent versions of the arguments in eq(6-17, 18) for the diffractogram image and for the template:

\[ \delta = 2(ar^2 + br^2 \cos(2\theta) + c r^4)(\text{image}) \]

\[ \varepsilon = 2(a'r^2 + b'r^2 \cos(2\theta) + c'r^4)(\text{template}) \]

\[ \text{eq(6-60, 61)} \]

The bracketed term in the correlation integral, eq(6-59), may now be written in the form

\[ (\ldots) = (1 + \gamma \cos(\delta))(1 + \xi_3) - (1 + \gamma' \cos(\varepsilon)) \]

\[ \text{eq(6-62)} \]
using eq(6-60) and eq(6-61) for \( f_2 \) and \( f_3 \). If \( \gamma = \gamma' \), i.e. the estimate of the fringe modulation is correct, squaring this expression gives:

\[
(\ldots)^2 = \gamma^2 (\cos \delta - \cos \epsilon)^2 \\
+ 2\gamma \xi_3 [ (\cos \delta - \cos \epsilon)(1 + \gamma \cos \delta) ] \\
+ \xi_3^2 (1 + \gamma \cos \delta)^2
\]

\text{eq}(6-63)

The expectation value of \( C \) therefore becomes:

\[
\overline{C} = \pi R^2 f_1^2 [\gamma^2 (\cos \delta - \cos \epsilon)^2 + \sigma_3^2 (1 + \gamma^2 / 2)]
\]

\text{eq}(6-64)

At the minimum, where \( \delta \) and \( \epsilon \) are identical, it becomes:

\[
\overline{C} = \pi R^2 f_1^2 \sigma_3^2 (1 + \gamma^2 / 2)
\]

\text{eq}(6-65)

Since we normalise the data such that \( \pi R^2 f_0^2 = 1 \), the expectation value of \( C \) for the normalised data becomes:

\[
\overline{C} = \left( \frac{f_1^2}{f_0^2} \right) \sigma_3^2 (1 + \gamma^2 / 2)
\]

\text{eq}(6-66)

However from eq(6-45), \( f_0^2 = f_1^2 (1 + \sigma_3^2) (1 + \gamma^2 / 2) \). Hence:

\[
\overline{C} = \frac{\sigma_3^2}{1 + \sigma_3^2}
\]

\text{eq}(6-67)

Using the value for the variance of 1.25 (given in section 6.7.2.2) gives \( \overline{C}_{\text{min}} = 0.55 \).

Figure 6-15 shows the experimental value to be 0.51.

6.7.5.1 Estimating the Shape of the Correlation Minimum in \( A_1 \) and \( C_1 \) Space

We now estimate the shape of the correlation minima generated by the template matching operation. The integrand in the correlation integral is:
\((....)^2 = \gamma^2 (\cos \delta - \cos \epsilon)^2 + \xi^2 + 2\gamma\xi [\cos \delta - \cos \epsilon + \gamma (\cos^2 \delta - 2\cos \delta \cos \epsilon)] + \gamma^2 \xi^2 (2\cos \delta + \gamma \cos^2 \delta)\)

If we assume for the moment that the noise terms are all zero then \((....)^2\) reduces to:

\[(....)^2 = \gamma^2 (\cos \delta - \cos \epsilon)^2 = \gamma^2 (\cos^2 \delta + \cos^2 \epsilon) - 2\gamma^2 \cos \delta \cos \epsilon\]

The integral of the first bracketed term is a constant, independent of the particular shape of the template, which evaluates to \(\gamma^2\), however the cross-product term does depend on the difference between the template and the image. Since

\[2\cos \delta \cos \epsilon = \cos(\delta - \epsilon) + \cos(\delta + \epsilon)\]

If we let \(a' = a + \Delta a\), \(b' = b + \Delta b\) and \(c' = c + \Delta c\), this gives

\[\delta + \epsilon = 4r^2[a + \Delta a/2] + 4r^2 \cos(2\theta)[b + \Delta b/2] + 4r^4[c + \Delta c/2]\]

and

\[\delta - \epsilon = 2r^2[-\Delta a] + 2r^2 \cos(2\theta)[-\Delta b] + 4r^4[-\Delta c]\]

if we assume for now that \(\Delta c = 0\), i.e. our estimate of the spherical aberration is correct, the term in \(r^4\) in the second equation disappears.

\[\delta - \epsilon = 2r^2[\Delta a + \Delta b \cos(2\theta)]\]

and

\[\delta + \epsilon = 4ar^4 + 4br^2 \cos(2\theta) + 4r^4 c + 2r^2[\Delta a + \Delta b \cos(2\theta)]\]

\[\approx 2\delta\]
Hence the equation for the correlation integral, eq(6-70), can be written in the form

\[ I = \text{constant} - \frac{\gamma^2}{\pi R^2} \int \int \cos(2r^2[\Delta a + \Delta b \cos(2\theta)])rdrd\theta \]  

eq(6-75)

We now evaluate the integral. Setting \( \rho = r^2 \) and \( \psi = 2\theta \), eq(6-75) becomes (ignoring the constant):

\[ I = \frac{\gamma^2}{4\pi R^2} \int_0^{4\pi} \int_0^{R^2} \cos(2\rho[\Delta a + \Delta b \cos\psi])d\rho d\psi \]  

eq(6-76)

Integrating over \( \rho \) yields:

\[ I = \frac{\gamma^2}{4\pi} \int_0^{\pi} \frac{1}{2R^2(\Delta a + \Delta b \cos\psi)} \sin(2R^2[\Delta a + \Delta b \cos\psi])d\psi \]  

eq(6-77)

The integrand is a sinc function. Since we are concerned only with the region where \( \Delta a/a \ll 1 \) and \( \Delta b/b \ll 1 \), in the region of the correlation peak, the modulus of the argument of the sinc function will be much less than \( \pi \). The integral can therefore be approximated by:

\[ I = \gamma^2 \int_0^{\pi} \left[ 1 - \frac{4R^4(\Delta a + \Delta b \cos\psi)^2}{3!} \right]d\psi \]

\[ = \gamma^2 \left[ 1 - \frac{R^4}{3}(2\Delta a^2 + \Delta b^2) \right] \]  

eq(6-78)

Since:

\[ a = \frac{C_1 \lambda \pi}{2}, b = \frac{A_1 \lambda \pi}{2} \]  

eq(6-79)

This yields:
Figure 6-14 Actual correlation minimum for the diffractogram shown in Figure 6-4 in $\Lambda_1$ and $C_1$ space.

Where $\Delta C_1 = C_1 - C_{1\text{min}}$ and $\Delta A_1 = A_1 - A_{1\text{min}}$ are the distances from the correlation minimum measured in $C_1$, $A_1$ space. Again, this has the quadratic variation with $\Delta C_1$ and $\Delta A_1$ to be expected for a correlation maximum or minimum. The validity of this function can by shown by comparing it to a correlation surface generated from a fit to a real diffractogram. Note that the integral is only valid in the region of the minima.

Figure 6-16 shows a contour plot for a wide range of values of defocus and astigmatism, Figure 6-14 shows a close up of the correlation minimum and Figure 6-15 shows cross sections through the minimum parallel to the two axes.

$$I = \gamma^2 \left[ 1 - \frac{\pi^2 R^4 A^2}{6} (\Delta C_1^2 + \frac{1}{2} \Delta A_1^2) \right]$$  

eq(6-80)
Figure 6-15 Cross section through the minimum shown in Figure 6-14

Figure 6-16 Contour plot of the correlation function $C(A_1, C_1)$ over a wide range of $A_1$ and $C_1$. The minimum corresponding to the actual values of astigmatism and defocus can be seen at $A_1 = 49\text{nm}$, $C_1 = 43\text{nm}$. This corresponds to the value of $A_1$ and $C_1$ estimated using manual fitting of the diffractogram. Note that this plot only shows one quadrant of the correlation surface, where $A_1$ and $C_1$ are both positive. The surface is actually symmetrical about the two axes (so only one quadrant is shown), and $a$ priori information about the beam-tilt must be used in conjunction with equations (6-2 to 6-7) to resolve the ambiguity in the sign of $A_1$ and $C_1$. 
Fitting a quadratic through the minimum to the two primary cross sections of the empirical data of the form

\[
C(\Delta C_1) = \alpha_{\text{defocus}} (\Delta C_1)^2 + \beta_{\text{defocus}} (\Delta C_1) + \chi_{\text{defocus}}
\]

and

\[
C(\Delta A_t) = \alpha_{\text{astig}} (\Delta A_t)^2 + \beta_{\text{astig}} (\Delta A_t) + \chi_{\text{astig}}
\]

yield \( \alpha_{\text{defocus}} = 6 \times 10^{14} \) and \( \alpha_{\text{defocus}} = 1.2 \times 10^{15} \).

From eq(6-71).

\[
\alpha_{\text{astig}} = \frac{1}{12} \gamma^2 \pi^2 R^4 \lambda^2
\]

\[
\alpha_{\text{defocus}} = \frac{1}{6} \gamma^2 \pi^2 R^4 \lambda^2
\]

A value for \( R \) of \( 3.5 \times 10^9 \text{m}^{-1} \) gives the predicted values of \( \alpha_{\text{defocus}} = 1.2 \times 10^{15} \) and \( \alpha_{\text{defocus}} = 6 \times 10^{14} \), corresponding to the empirical results. This encloses only around 60% of the energy. However these expressions again have \( R^4 \) dependency, and consequently are very sensitive to errors in the estimated value of \( R \).

The radius over which the energy of the diffractogram is contained can be estimated by integrating the energy over a variety of radii and plotting the results. The graph in Figure 6-17 shows the variation in energy for the diffractogram used in the analysis. As can be seen the energy approaches 100% asymptotically, making the estimate of \( R \) quite sensitive to the choice of \( E_{\text{max}} \), the proportion of the total energy which we consider to contain all the diffractogram information. Setting \( E_{\text{max}} \) to 90% yields a value of 5.3nm\(^{-1}\). This appears to be a reasonable approximation, based on the values derived from eq(6-83,6-84) and the region of the diffractogram which appears to the human eye to contain the information.
Figure 6-17 Variation in enclosed energy as a function of the radius of integration. 90% of the energy is enclosed at a maximum spatial frequency of about 5.3nm\(^{-1}\), 99% is enclosed at 7.7nm\(^{-1}\).

6.7.6 Estimating the Errors in \(A_1\), \(C_1\) and \(C_3\).

Having found the shape of the correlation minimum, and confirmed that our model matches the empirical data, we can analyse the affect of noise on its accuracy in predicting the aberrations. Consider again the cross-term in the integrand in the correlation minima from eq(6-63):

\[
2(1 + \xi_3)(1 + \gamma \cos \delta)(1 + \gamma \cos \epsilon)
\]

\text{eq(6-85)}

This varies considerably from the equation for the integrand for the determination of \(\phi_{22}\), given in eq(6-27), as there is now only one noise term, which behaves somewhat differently from the previous analysis. This term determines the position and shape of the correlation minimum. We have already shown it to have a quadratic shape in the noise-free case. However, we will show that the presence of noise produces a shift in the position of the minimum in this function from one image sample to another. We remind the reader that the ‘noise’ is the Fourier transform of the amorphous carbon film, and its structure is fixed (for a given image), but unpredictable from one image to another. Different images will therefore have different noise patterns, even though they may have the same aberration coefficients. Given a particular image, with a fixed but random ‘noise’ pattern, the shape of the minimum will be a quadratic function of the form:
\[ f(x) = 2jx^2 + 2kx + 2l \]

where \( x \) denotes a small shift, \( \Delta a \) or \( \Delta b \), in either of the aberration coefficients \( a \) or \( b \) from their 'true' values \( a_0, b_0 \). We assume here that only one or other of \( \Delta a \) or \( \Delta b \) is non-zero, in order to simplify the analysis. The coefficients \( j, k, \) and \( l \) in eq(6-86) are functions of the true aberrations and of the noise, and will be determined below. The position of the minimum, \( x_{\text{min}} \), in eq(6-86) is determined by the relation:
\[
\frac{\partial f}{\partial x} = 0
\]
eq(6-87)

i.e. when:

\[
x_{\text{min}} = \frac{k}{2j}
\]
eq(6-88)

If the noise is zero, then \( k = 0, \) and \( x_{\text{min}} = 0. \) However when the noise terms are included, then \( j \) and \( k \) will vary from one image to another, resulting in an error in the estimation of the position of the minimum, and hence of the particular image aberrations. Thus by determining the variance in the coefficients, \( j \) and \( k, \) (due to the noise terms) it is possible to estimate the error in \( x_{\text{min}}, \) and hence the errors in the aberration coefficients using the relationship [55]:

\[
\sigma^2_{x_{\text{min}}} = \frac{\sigma_j^2 + \sigma_k^2 \frac{k_0^2}{4j_0^2}}{4j_0^2}
\]
eq(6-89)

where \( \sigma_{x_{\text{min}}}^2 \) is the variance in the minima position and \( \sigma_j^2, \sigma_k^2 \) and \( j_0, k_0 \) are the variances and means of \( j \) and \( k \) respectively.

We now return to eq(6-85), and evaluate \( j \) and \( k, \) and their variances. The integral becomes (after normalisation):

\[
I = -2 \frac{f^2}{\pi R^2 f_0^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (1 + \xi_3)(1 + \gamma \cos \delta)(1 + \gamma \cos \epsilon) r dr d\theta
\]
eq(6-90)

where \( C \) is the circular region of radius \( R. \) We introduce the variable \( r, \) indicating the vector position in \( r, \theta \) space, and make the substitutions:
\[ g(r) = 1 + \gamma \cos \delta \]
\[ B = \frac{f_i^2}{\pi R^2 f_0^2} = \frac{1}{\pi R^2 (1 + \sigma_i^2)(1 + \gamma^2 / 2)} \]

eq(6-91,92)

where \( g(r) \) is the underlying (noise-free) scene shape and \( r \) is the vector position of a point, \((r,0)\). Expanding the template term \( 1 + \gamma \cos \epsilon \) around the point \( x = 0 \) (using \( x \) to denote either \( \Delta a \) or \( \Delta b \)), and using the small angle approximations, \( \sin x = x, \cos x = 1 - x^2/2 \), yield an expression of the form:

\[ 1 + \gamma \cos \epsilon = x^2 M(r) + xN(r) + P(r) \]

eq(6-93)

where \( M(r), N(r) \) and \( P(r) \) are the Taylor coefficients, which depend on the image aberrations, and are to be determined. Eq(6-90) now becomes:

\[ I = -2B \int_c (1 + \xi_j g(r) \left\{ x^2 M(r) + xN(r) + P(r) \right\} dr \]

eq(6-94)

This expression can be separated into a series of polynomial terms in \( x \), which is equivalent to the expression given in eq(6-86) with:

\[ j = -2B \left[ \int_c \xi_j g(r) M(r) dr + \int_c g(r) M(r) dr \right] \]

\[ k = -2B \left[ \int_c \xi_j g(r) N(r) dr + \int_c g(r) N(r) dr \right] \]

\[ l = -2B \left[ \int_c \xi_j g(r) P(r) dr + \int_c g(r) P(r) dr \right] \]

eq(6-95,96,97)

The constant term, \( l \), does not affect the position of the minimum given by eq(6-88), and can be ignored. The expectation value of \( j \) is:

\[ \langle j \rangle = -2B \int_c g(r) M(r) dr \]

eq(6-98)
because $\langle \xi_3 \rangle$, the expectation value of $\xi_3$, is zero. The variance of $k$, $\text{var}(k) = \langle k^2 \rangle - \langle k \rangle^2$, is:

$$
\text{var}(k) = \left\langle \left[ -2B \int_{\mathcal{C}} \xi_3 g(r)N(r)dr + B \int_{\mathcal{C}} g(r)N(r)dr \right] \right\rangle^2,
$$

which is equal to:

$$
\text{var}(k) = \left\langle \left[ -2B \int_{\mathcal{C}} \xi_3 g(r)N(r)dr \right] \right\rangle
$$

as the expectation of the noise cross product term is zero. Eq(6-100) can be written in the form:

$$
\text{var}(k) = 4B^2 \left( \int_{\mathcal{C}} \xi_3(y) \xi_3(z) g(r)N(r)g(s)N(s)drds \right)
$$

where $\xi_3(y)$ and $\xi_3(z)$ denote the values of the noise field at points $y$ and $z$, given by the vectors $r$ and $s$. At this point it is necessary to consider the behaviour of the function when it is evaluated as a discrete summation by the algorithm. As a discrete summation, eq(6-101) takes the form:

$$
\text{var}(k) = 4B^2 p^2 \sum_{r=1}^{A} \xi_r g_r, N_r \sum_{s=1}^{A} \xi_s g_s, N_s
$$

eq(6-102)
where $g_r$ and $N_r$ denote the value of the functions $g(r)$ and $N(r)$ at the discrete pixel position $r$ corresponding to the vector position $\mathbf{r}$. $A$ is the number of pixels over which the summation is performed and $p$ is the size of each pixel, $p = \pi R^2 / A$.

If $\xi_3$ is uncorrelated, as we have assumed, then $E[\xi_3(r), \xi_3(s)] = 0$ for $r \neq s$, and the summation tends to zero, but when $r = s$ then:

$$E[\xi_3(r), \xi_3(s)] = \sigma_3^2, \quad r = s,$$

where $\sigma_3^2$ is the variance of the noise in the scene image. Thus eq(6-102) can be written as:

$$\text{var}(k) = 4B^2 p^2 \sigma_3^2 A(gN)^2$$

eq(6-104)

where $(gN)^2$ is the mean value of $g_r^2 N_r^2$ and is equal to:

$$\frac{(gN)^2}{\pi R^2} = c \frac{\int g^2(r) N^2(r) dr}{\pi R^2}$$

eq(6-105)

Hence:

$$\text{var}(k) = \frac{4\sigma_3^2}{A(1 + \sigma_3^2)^2} \frac{\int g^2(r) N^2(r) dr}{\pi R^2}$$

eq(6-106)

Using the same argument:

$$\text{var}(j) = \frac{4\sigma_3^2}{A(1 + \sigma_3^2)^2} \frac{\int g^2(r) M^2(r) dr}{\pi R^2}$$

eq(6-107)

Combining these results with eq(6-89) enables the variance in the minimum position to be calculated.
The coefficients $M(r)$ and $N(r)$ can be determined by expanding $\cos \varepsilon$ around the position of the minimum. We first evaluate them for a change in $\Delta a$, and let $\Delta b$ and $\Delta c$ equal zero:

$$\cos \varepsilon = \cos \left\{ 2r^2(a + \Delta a) + 2r^2 \cos(2\theta)b + 2r^4c \right\}$$

$$= \cos(2r^2\Delta a) \cos \left\{ 2r^2a + 2r^2 \cos(2\theta)b + 2r^4c \right\}$$

$$- \sin(2r^2\Delta a) \sin \left\{ 2r^2a + 2r^2 \cos(2\theta)b + 2r^4c \right\}$$

\text{eq}(6-108)

Using the small angle approximations, $\sin x = x$, $\cos x = 1 - x^2/2$, yields:

$$\cos \varepsilon = \left[ 1 - (2r^2\Delta a)^2 / 2 \right] \cos \left\{ 2r^2a + 2r^2 \cos(2\theta)b + 2r^4c \right\}$$

$$- 2r^2\Delta a \sin \left\{ 2r^2a + 2r^2 \cos(2\theta)b + 2r^4c \right\}$$

\text{eq}(6-109)

Comparing with \text{eq}(6-94)\) therefore gives, for a small distance from the minimum in the $\Delta a$ direction:

$$M(r) = -2r^4\gamma \cos \delta$$

$$N(r) = 2r^2 \sin \delta$$

$$P(r) = 1 + \gamma \cos \delta$$

\text{eq}(6-110,111,112)

Therefore, using \text{eq}(6-98)\) and \text{eq}(6-106), <j> and var(k) can be determined, and hence, using \text{eq}(6-89)\) the variance in the position of the minimum, $\sigma_{x_{\text{min}}}^2$. 

166
From eq(6-98):

\[
\langle j \rangle = -2B \int g(r)M(r)dr
\]

\[
eq -\frac{4\gamma}{\pi R^2(1+\sigma_z^2)(1+\gamma^2/2)} \int_0^\infty \left[ \cos \delta + \gamma \cos^2 \delta \right] r^3 drd\theta
\]

eq(6-113)

and noting that the \( \cos \) term averages to zero, while the \( \cos^2 \) term averages to \( \frac{1}{2} \), we obtain:

\[
\langle j \rangle \approx -\frac{2\gamma^2 R^4}{3(1+\sigma_z^2)(1+\gamma^2/2)}
\]

eq(6-114)

The noise-free expectation value of the quadratic component, \( \langle j \rangle \), is the same as \( \alpha_{\text{defocus}} \), which was evaluated in eq(6-84).

Similarly, the expectation value of \( k \) is given by:

\[
\langle k \rangle = -2B \int g(r)N(r)dr
\]

\[
eq -\frac{4\gamma}{\pi R^2(1+\sigma_z^2)(1+\gamma^2/2)} \int_0^\infty \left[ 1 + \gamma \cos^2 \delta \right] \sin \delta \ r^3 drd\theta
\]

eq(6-115)

The sine function in eq(6-115) integrates to zero, giving \( \langle k \rangle = 0 \), indicating that we expect no systematic error from eq(6-88), and hence the value of \( \text{var}(j) \) does not affect the error estimate. Thus \( \text{var}(j) \) does not need to be evaluated. However:

\[
\text{var}(k) = \frac{4\sigma_z^2}{A(1+\sigma_z^2)^2(1+\gamma^2/2)^2\pi R^2} \int g^2(r)N^2(r)dr
\]

\[
= \frac{4\gamma^2 \sigma_z^2}{A(1+\sigma_z^2)^2(1+\gamma^2/2)^2\pi R^2} \int_0^\infty \left[ 1 + \gamma \cos \delta \right]^2 \sin^2 \delta \ r^5 drd\theta
\]

eq(6-116)

Again, we approximate the sinusoid terms by their average. Thus eq(6-116) can be approximated by:
\[ \text{var}(k) = \frac{\sigma_3^2 \gamma^2 [4 + 3\gamma^2] R^4}{6A(1 + \sigma_3^2)^2 (1 + \gamma^2 / 2)^2} \]  
\text{eq}(6-117) 

and:

\[ \sigma_k = \sqrt{\frac{\sigma_3^2 \gamma^2 [4 + 3\gamma^2] R^4}{6A(1 + \sigma_3^2)^2 (1 + \gamma^2 / 2)^2}} \]  
\text{eq}(6-118) 

Given \( \langle k \rangle = 0 \), as the sin term integrates to zero, this yields from eq(6-89):

\[ \sigma_{\text{x}_\text{mn}} = \frac{\sigma_k}{2 \langle j \rangle} = \frac{\sigma_3}{4\gamma R^2} \sqrt{\frac{(4 + 3\gamma^2)}{6A}} \]  
\text{eq}(6-119) 

The relationship between the minimum in \( \Delta a \) space and the error in the actual aberration coefficient was given in eq(6-79), yielding:

\[ \sigma_{C_1} = \frac{2\sigma_{\text{x}_\text{mn}}}{\pi \lambda} \]  
\text{eq}(6-120) 

Using the values for \( \gamma \) and \( \sigma \) estimated in section 6.6.2.2, (\( \gamma = 0.65 \) and \( \sigma = 1.15 \)) and given a sampling frequency in the images of 0.0333nm/pixel, \( R = 5 \times 10^9 \) cycles/m and \( A \approx 2 \times 10^5 \) pixels, yields an estimate of the standard deviation of the error in the estimate of \( \sigma_{C_1} \) of:

\[ \sigma_{C_1} = 0.03 \text{ nm} \]

This is of the order of 0.1% on the estimated value of \( C_1 \) of 43nm.

Performing the same analysis, but varying \( \Delta b \) with \( \Delta a = 0 \) (to determine the equivalent error in the estimate of the apparent astigmatism, \( A_1 \)):
\[
\cos \varepsilon = \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)[b + \Delta b] + 2r^4 c \right\} \\
= \cos(2r^2 \cos(2\theta)\Delta b) \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
- \sin(2r^2 \cos(2\theta)\Delta b) \sin \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
\text{eq}(6\text{-}121)
\]

Again, using the small angle approximations, \(\sin x = x\), \(\cos x = 1 - x^2/2\), yields:

\[
\cos \varepsilon = \left[ 1 - (2r^2 \cos(2\theta)\Delta b)^2 / 2 \right] \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
- 2r^2 \cos(2\theta)\Delta b \sin \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
\text{eq}(6\text{-}122)
\]

Comparing with eq(6\text{-}93) for a small distance from the minimum in the \(\Delta b\) direction:

\[
M(r) = -2r^4 \gamma \cos^2(2\theta) \cos \delta \\
N(r) = 2r^2 \gamma \cos(2\theta) \sin \delta \\
P(r) = 1 + \gamma \cos \delta \\
\text{eq}(6\text{-}123,124,125)
\]

which, when integrated over a circle reduce \(\langle j \rangle\) by a factor of 2 and \(\text{var}(k)\) by a factor of 4. Hence, using the same values for \(\gamma\), \(\sigma\) and \(R\), yields an estimate of the standard deviation of the error in the estimate of \(\sigma_{A_i}\) of:

\[
\sigma_{A_i} = 0.03 \text{ nm}
\]

6.7.6.1 Estimating the Errors in \(C_3\)

So far, we have assumed that the manufacturer’s estimate of \(C_3\) is correct. Using the same analysis as was used for estimating the errors in \(C_1\) and \(A_1\), we can now relax this assumption and estimate the probable error in \(C_3\) if template matching is used to determine its value.

If we set \(\Delta a\) and \(\Delta b\) to zero, and let \(c = c + \Delta c\), eq(6-108) now approximates to:

\[
\cos \varepsilon = \left[ 1 - (r^4 \Delta c)^2 / 2 \right] \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
- 2r^4 \Delta c \sin \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\} \\
\text{eq}(6\text{-}126)
\]
This yields new expressions for $M(r)$, $N(r)$ and $P(r)$:

\[
M(r) = -2r^8 \gamma \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\}
\]

\[
N(r) = 2r^4 \gamma \sin \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\}
\]

\[
P(r) = 1 + \gamma \cos \left\{ 2r^2 a + 2r^2 \cos(2\theta)b + 2r^4 c \right\}
\]

eq(6-127)

Using these expressions, the expectation value of $j$, $<j>$, and var($k$) can be evaluated approximately as:

\[
<j> = -2B \int g(r)M(r)dr = -4B \int (1 + \gamma \cos \delta) \gamma \cos \delta \cdot r^9 dr d\theta
\]

\[
= \frac{2\gamma^2 R^8}{5(1 + \sigma_3^2)(1 + \gamma^2 / 2)}
\]

eq(6-128)

and:

\[
\text{var}(k) = \frac{4\sigma_3^2}{A(1 + \sigma_3^2)^2(1 + \gamma^2 / 2)^2 \pi R^2} \int g^2(r)N^2(r)dr
\]

\[
= \frac{\sigma_3^2 \gamma^2 [8 + 6\gamma^2] R^8}{5A(1 + \sigma_3^2)^2(1 + \gamma^2 / 2)^2}
\]

eq(6-129)

Substituting these values into eq(6-89) gives:

\[
\sigma_{x\text{min}} = \frac{\sigma_3}{\gamma R^4} \sqrt{\frac{5(4 + 3\gamma^2)}{2A}}
\]

eq(6-130)

Using the same values for $\gamma$, $\sigma$ and $R$, yields an estimate of the standard deviation of the error in the estimate of $\sigma_{c_3}$ of:

\[
\sigma_{c_3} = 3 \mu m
\]

This is gives a relative error of approximately 0.3% on the manufacturer’s value for $C_3$ of 1.1mm. Table 6-2 summarises the values of all the aberrations analysed and the predicted absolute and relative errors.
### Table 6-2. Summary of the values of the aberrations for the diffractogram used and their estimated absolute and relative errors.

<table>
<thead>
<tr>
<th>Aberration</th>
<th>Value</th>
<th>Estimated Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defocus, $C_1$</td>
<td>43nm</td>
<td>0.03nm (-0.1%)</td>
</tr>
<tr>
<td>Astigmatism, $A_1$</td>
<td>49nm</td>
<td>0.03nm (-0.1%)</td>
</tr>
<tr>
<td>Spherical aberration, $C_3$</td>
<td>1.1mm</td>
<td>3μm (-0.3%)</td>
</tr>
</tbody>
</table>

Comparing this to real data is difficult, as ground truth values of the aberrations are not known to within this degree of accuracy. Generating synthetic data with the same noise standard deviation yielded an ‘experimental’ error of ±0.05nm for $A_1$ and $C_1$, which compares reasonably well with the predicted value of $\sigma_{C_1}$ and $\sigma_{A_1}$. It is anticipated that the errors in fits to the real data will be higher, as the noise is not entirely multiplicative (there is still some noise at the bottom of the fringes). The estimate of the error in $C_3$ of 0.3% however is significantly lower than that observed in independent experimental measurements. We have assumed that our value for the scale of the diffractogram, and hence the sampling interval is correct. In practice it is difficult to determine the sampling interval with a high degree of accuracy, and a small error in this estimate could easily accommodate the errors observed in the aberrations in experimental measurements. We have also assumed in this analysis that artifacts due to crystalline inclusions have been completely removed, which may not always be the case. The largest error found was in the determination of $\phi_{22}$, which could result in a significant error in the estimates of $A_{1X}$ and $A_{1Y}$, the $x$ and $y$ components of the astigmatism. However, these could be reduced by repeating the correlation operation in the region of the minimum, using synthetic templates with $\phi_{22}$ varying by around 50mrad either side of the value estimated by module 3.

### 6.7.6.2 Effect of Errors on the Measurements of the True Aberrations

The errors that have been predicted in the previous section are the errors in the apparent aberrations. These apparent aberrations are related to the true lens aberrations according to eq(6-2,3,4,5,6,7), reprinted here:
From eq(6-7) $C'_3 = C_3$. Therefore the expected error in the measurement of the true spherical aberration is $\sigma_{c_3} = 3\mu$m. However, determining the other true aberrations requires the beam-tilt to be measured. If the measurement of beam-tilt is significantly less accurate than the predicted errors in the determination of the apparent aberrations, which are around 0.1\%, then it is the beam-tilt errors which will dominate the errors in the measurement of the true aberrations. In practice it is possible to refine the values of the beam-tilt [99]. Here only the errors in the apparent aberration coefficients are considered, an estimate of the errors in the beam-tilt, determined by performing a fit to successively more diffractograms is given in [76]. Given sufficiently accurate measurement of the beam-tilt however, these results show that image interpretation at the 0.13nm level is possible [75]. Unfortunately, the model presented in this chapter does not allow for $C_3$ and $C_1$ to be varied simultaneously. It has been found using manual fitting techniques that there is a strong interdependence between the estimates of $C_3$ and $C_1$, and a variation in one can be accommodated by a different value for the other aberration [75]. An analysis of the effects of errors being propagated through eq(6-2,3,4,5,6,7) is given in [95].

6.8 Summary

This chapter has described an analytical model of a lens aberration determination algorithm for a transmission electron microscope. The model predicted that under typical imaging conditions in the TEM, a template matching based algorithm would measure the apparent aberrations with a standard deviation in the error of around 0.1 on defocus and astigmatism values of 43nm and 49nm, and of 0.3\% on a spherical aberration value of 1.1mm. However, the accuracy with which most of the true aberrations could be deduced from the measured apparent aberrations was shown to depend critically on the accuracy with which the beam-tilt and sampling interval could
be measured. The sampling interval measurement could prove to be the dominant source of errors if it cannot be determined with a comparable degree of accuracy to that with which the apparent aberrations have been measured. Knowledge of the beam-tilt and sampling interval errors would then enable the full system performance to be predicted.
Chapter 7: Drum Location

This chapter describes the analysis of a vision system which is used by an automated handling process in a nuclear fuel repository to locate the position of drums of fuel. The problem is well constrained, and an analytical model of the vision system behaviour is developed. However a final performance figure for the IP system is not evaluated, as a full probability distribution of the operating conditions was not available. The problem falls into category three, the performance is likely to be dominated by non-imaging factors. Using reasonable estimates for the imaging parameters, it appears that the IP system will prove reliable. However errors are likely to be caused by other factors, which are external to the imaging and vision system, and difficult to predict.

7.1 Background

One of the stages in the reprocessing and reuse of nuclear fuel is the storage of reprocessed Uranium Oxide (UO$_3$). The oxide powder is stored in 200kg stainless steel drums, resembling milk churns, which are stored in a specially designed and shielded warehouse. Handling of these drums is performed by Self-Guided Vehicles (SGVs) which place and retrieve the drums without direct operator control. This minimises the radiation exposure of the operators. The SGVs use a video imaging system to locate the correct position for placing and picking up the drums [100].

7.1.1 Operation of the Storage Plant

The drums arrive at the facility on trucks and are stacked onto a conveyer. A lift then transports the drums to the correct floor for storage and the SGV takes the drum to the required position. The drums are stored in ‘bins’, each 3 drums wide, 4 high and 12 deep. Each ‘bin’ consists of a designated area of floor space, marked by a line of twelve spots leading away from the wall. The drums are placed one directly on top of each spot and one each side. Other drums are then placed on top of these, and then another row is started in front of this: see Figure 7-1.
There are a total of 200 bins in the current storage facility, giving it a capacity of 28,800 drums. The storage is not permanent. Although not currently used as a fuel, the reprocessed UO₃ is intended for shipping to customers to be used as a replacement for UO₃ from mined uranium. The locations of a customer’s drums are logged and the SGVs are required to be able to retrieve the drums as well as deposit them. Due to the design of the bins, the SGV can only retrieve drums at the top of the front row.

7.1.2 Operation of the Self Guided Vehicles

The SGVs are based on a standard design, with a special actuator and vision system added. There are two position location devices. The main system uses a rotating laser scanner which determines the SGV’s position by using ‘bar-code’ markers on the walls. This is backed up by a dead-reckoner which logs the distance and direction travelled based on the rotation of the wheels. The SGV is moved into the approximate position to pick up or set-down the drum by the laser positioning system, with the dead-reckoner acting as a back-up. The largest positional error registered during testing was 19mm relying on the dead-reckoner alone. Once in position, a CCD camera on the grabber takes a snapshot, accompanied by a flash from two spotlights either side of the camera. This single image is processed to provide precise information on the position of the drum or the marker spot on the floor for subsequent control of the actuator.
The actuator has a three-point grabber which can be moved vertically, sideways, and for-and-aft. The grabber closes around the rim of the drum. The mechanical design of the grabber is such that it can only release the drum when there is no weight on the clamps, i.e. when the drum is placed on a supporting surface. This prevents the SGV from dropping the drum in the event of a failure of the control system. As a further precaution, tilt switches in the grabber prevent the clamps from releasing if the drum is not vertical. There is also a second CCD on the SGV which relays a side view of the operation to a human operator. Pick-up or set-down is not performed until the operator has confirmed that the SGV is in position.

All signals from the cameras and other sensors are transmitted back to a base station via a radio transmitter and 'leaky feeders'. The image processing computation to determine the exact location of the drum is performed at the base station, out of the storage area, and the control signals are transmitted back to the SGV. This is to keep the computers out of the controlled area, to minimise the radiation damage to the electronics.

The SGV is shown in Figure 7-2 performing tests at the plant. The grabber can be seen on the end of the actuator arm with the camera and lights above it. The laser scanner is visible on top of the main body of the SGV, to the left-hand side of the image.
Figure 7-2 SGV with test drums.

## 7.2 Description of the Image Processing

The image processing task is to locate the centre of an ellipse appearing somewhere in the field of view. This represents the cap of the drum below or the circular marker on the floor, as seen from the camera on the actuator arm. Two views taken from the camera are shown below.

![Image of drums](image.png)

Figure 7-3 UO₃ Drums. The cap, clearly visible in the centre of the images, was chosen from a series of designs to give maximum contrast separation between 'light' and 'dark' areas. The outer ring of the cap is painted black, and the centre is left as stainless steel.

### 7.2.1 Algorithm Description

The image processing algorithm produces a histogram of the grey levels in the captured image. A threshold intensity, T, is then chosen and the image thresholded at this intensity. Using a seed pixel in the centre of the image, which is assumed to already have intensity, $I > T$, a linking procedure is now performed such that a labeled image $L(x,y)$ is produced according to:

$$
l_{x,y} = \begin{cases} 
1 & I_{x,y} > T \land L_{x-1,y-1} = 1 \lor L_{x+1,y-1} = 1 \lor L_{x-1,y+1} = 1 \lor L_{x+1,y+1} = 1 \\
0 & \text{otherwise} 
\end{cases}
$$

\text{eq}(7-1)
i.e. if any pixel has intensity \( I > T \), and at least one of its eight-connected neighbours has already been classified as part of the seed pixel region, it too is classified as part of the region.

The area of this segmented region is now calculated. If it is less than some value, \( A_{\text{min}} \), equal to the expected minimum area of the centre reflector in the image, then the threshold, \( T \), is reduced by a pre-determined amount. The process is then repeated until a satisfactory match is found with a linked area \( \geq A_{\text{min}} \). This area is then assumed to be the top of the cap and its centre of intensity is found and taken as the position of the drum [101].

![Figure 7-4 A block diagram of the image processing system](image)

### 7.3 Problem Analysis

#### 7.3.1 Parameters Affecting Performance

After discussing the operation of the image processing system with the designer and viewing sample images, the parameters affecting performance were estimated and ranked as shown in Table 7-1.

<table>
<thead>
<tr>
<th>Image Parameter</th>
<th>Rank severity</th>
<th>Rank Occurrence</th>
<th>Effect on Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambient lighting</td>
<td>7</td>
<td>5</td>
<td>contrast falls, illumination profile changes</td>
</tr>
<tr>
<td>spotlight failure</td>
<td>6</td>
<td>7</td>
<td>contrast falls, illumination profile changes</td>
</tr>
<tr>
<td>paint deterioration</td>
<td>11</td>
<td>6</td>
<td>reflection profile changes</td>
</tr>
<tr>
<td>signal degradation during transmission</td>
<td>1</td>
<td>8</td>
<td>unknown</td>
</tr>
<tr>
<td>signal degradation due to radiation-induced deterioration of the CCD</td>
<td>2</td>
<td>9</td>
<td>unknown</td>
</tr>
</tbody>
</table>
Table 7-1 Parameters affecting the performance of the IP system on the SGV. The parameters have been ranked according to their estimated severity and their probability of having a value which will be detrimental to the algorithm performance.

7.3.2 Performance Measures

The performance of the image processing algorithm can be determined by its likelihood of locating the centre of the drum cap to within ±3mm in any direction. This is the maximum tolerance with which the grabber can operate.

7.3.3 Problem Classification

The imaging conditions are quite well constrained due to the controlled environment of the nuclear waste store. Most of the parameters can be reasonably well quantified or modelled. However several external parameters, such as multi-path effects, radiation induced damage and signal degradation, which rank highly in the list of severity, are very difficult to quantify. The problem is thus a category three problem, one which is not completely characterisable, due to non-imaging variables dominating performance.

7.4 System Analysis

7.4.1 Algorithm Modularisation

Although the algorithm actually operates as an iterative loop as it searches for an area with intensity, $I \geq T$ and area $\geq A_{\text{min}}$, it is possible to simplify the modularisation of
the algorithm by considering just the final result of the iterative procedure, which is then passed to the last module. Thus we can assume that the image is thresholded once, then linked such that the area linked, $A_{\text{link}}$, is confined to $A_{\min} \leq A_{\text{link}} \leq A_{\min} + \delta$ where $\delta$ is some small value which is the result of the finite step size of the changes in threshold value. As only the output image is relevant to the following processing stage, the intermediate steps where the final threshold is found do not need to be analysed. The algorithm now reduces to a simpler form as shown in Figure 7-5.

![Figure 7-5 Simplified block diagram of the image processing system. The iterative loop no longer needs to be considered.](image)

7.4.2 Interaction and Propagation of Parameters

All the external parameters interact with Module 1 to determine the quality and position of the thresholded region. The performance of Module 2 is only determined by the thresholded image. The ultimate performance measure is the probable distribution of the error in the $x$ and $y$ coordinates, and its probability of being greater than 3mm. This is affected by the probability distribution of pixels which are part of the central cap region, being mislabelled as part of the background, and the probability of background pixels being mislabelled as part of the central cap region. This is likely to occur in one of two ways. Either the wrong region will be linked, if the seed pixel is not part of the ladle cap, or more likely, the linked region will not match the true region closely enough for centre of gravity calculation to produce a result within tolerance. The interaction and propagation of parameters is as follows:

Module 1. Thresholder - Illumination profile, noise, $\sigma$; area obscured by dirt and dust, $\rho$; area with effective obscuration due to paint deterioration, $\rho$; background grey level $\beta$, expected area of drum cap $A_{\min}$, dependent on the SGV positioning errors; drum cap true size $A$; error in SGV position resulting in error in seed pixel position $\Delta u, \Delta v$. 

180
Module 2. Linker - Probability distribution of true pixels being above threshold, \( m(x,y) \); probability distribution of non-cap pixels being above threshold, \( n(x,y) \).

Module 3. Centre finding - Probability distribution of true cap pixels being correctly linked, \( p(x,y) \); probability distribution of non-cap pixels being incorrectly linked, \( q(x,y) \); probability of incorrect region being linked, \( \varepsilon \).

The full parameter propagation diagram is shown in Figure 7-6.

**Figure 7-6 Parameter propagation for the drum location system**

7.4.3 Model of Imaging Conditions

The effects of lighting conditions are not readily quantifiable, and not enough data is available from the actual test site for an empirical distribution of the operating conditions to be generated. Consequently it is necessary to model the imaging conditions to estimate the effect of lighting on the illumination profile.

If the apparent grey-level at any point in a surface in an image \( I(x,y) \) is assumed to be a function of illumination at that point \( c(x,y) \), reflectance \( r(x,y) \), the angle of illumination \( \theta_c(x,y) \) and the viewing angle \( \theta_v(x,y) \), then

\[
I(x,y) = f[c(x,y), r(x,y), \theta_c(x,y), \theta_v(x,y)]
\]

\[\text{eq}(7-2)\]
The imaging conditions are also affected by the specific camera gain and other parameters. However as this data is unavailable, it will be assumed to have approximately the same characteristics as the camera used in the test rig used to establish the relationship between illumination profile and lighting conditions. As the information is extracted from a fairly narrow viewing angle, the effect of the viewing angle stays roughly constant and the effect of the illumination angle can be considered as part of the illumination profile. Thus

\[ I(x,y) = g[c(x,y), r(x,y)] \]

\[ \text{eq}(7-3) \]

A test rig was then set up with a camera poised above a drum cap with the same viewing and lighting geometry that would be experienced by the SGV in normal operation. The drum cap was cleaned to give a roughly constant value of \( r(x,y) \). (There were still some slight variations in the cap reflectance due to flaws in the surface texture). The ambient lighting and spotlights were then varied over a range approximating the likely operating conditions, and the effect on the image measured by fitting a bi-quadratic surface, \( h(x,y) = r(x,y)I(x,y) \) to the region of the image containing the central cap marker. This yielded a series of polynomial coefficients, \( h(x,y) = ax^2 + bx + cy^2 + dy + exy + f \), measuring the illumination at a point in the image in grey levels. It also yields a noise term, \( \sigma \), measuring the standard deviation in grey levels of the error between the fit and the actual illumination surface. Each of these fits quantitatively characterises one of the illumination conditions tested. The background grey level from the region surrounding the cap marker, \( \beta \), was also measured. A list of the illumination conditions and characteristic coefficients is given in table 7-2.

7.5 Performance Characterisation

7.5.1 Module 1: Thresholder

The datastream and performance parameter propagation for Module 1 is as follows:
Consider first the simplest failure mode, that is the wrong region is linked. This can only occur if the seed pixel is outside the centre circle in the cap image. Under these circumstances, the linking will ‘flood’ the background region, resulting in a spurious region whose centre will almost certainly be outside the minimum error of 2mm. i.e.

\[
\varepsilon = 0 \quad \sqrt{\Delta u^2 + \Delta v^2} \leq r
\]

\[
\varepsilon = 1 \quad \sqrt{\Delta u^2 + \Delta v^2} > r
\]

where \( r \) is the radius of the marker circle, \( r = 38\text{mm} \).

The second transfer function is the relationship between the illumination profile and the probability distribution. If we consider the model with the smoothly varying quadratic illumination profile, with a Gaussian noise component \( \zeta \) with variance \( \sigma^2 \):

\[
I(x, y) = s(x, y)\left[ax^2 + bx + cy^2 + dy + exy + f\right] + \zeta
\]

\[\text{eq}(7-6)\]

If we keep surface reflectance, \( s(x,y) \) constant (no dirt or paint damage) we can determine the probability of any particular pixel being above the threshold grey level, \( T \).
The probability, $m(x, y)$ that a pixel is below the threshold, $T$, is the probability that $\zeta > c(x, y) - T$. The noise $\zeta$ is assumed to be Gaussian distributed with variance $\sigma^2$, so that:

$$m(x, y) = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{c(x, y) - T}{\sigma \sqrt{2}} \right) \right)$$

**eq(7-7)**

where

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} \, dt$$

**eq(7-8)**

is the error function.

Similarly, the probability, $n$, of getting a spurious pixel is

$$n = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{T - \beta}{\sigma \sqrt{2}} \right) \right)$$

**eq(7-9)**

where $\beta$ is the background grey level. These functions are non-analytic, but readily evaluated for a given $c(x,y)$, $\beta$ and $T$.
The threshold, $T$, is the grey level at which $A_{\min} \leq A_{\text{link}} \leq A_{\min} + \delta$. $T$ cannot be determined explicitly, but is the value for which

\[ \int_{dA} p(x, y) = A_{\min} \]

**eq(7-10)**

We now let our surface reflectance, $r(x,y)$, vary by allowing some proportion, $\rho$, of $r(x,y)$, take a value, $c$, due to dust or paint damage. As we have no prior information on the likely reflectance, $c$, of any dust and or paint damage, we shall assume it results in a uniformly distributed grey level in $I(x,y)$. Thus

\[ p(c > T) = \frac{255 - T}{255} \]

**eq(7-11)**

Also, assuming the spatial distribution of the dirt or damage is random, independent and uniform, then the probability of any one pixel being missed is

\[ p(\text{missedpixel}) = \rho \frac{255 - T}{255} \]

**eq(7-12)**

and

\[ p(\text{spuriouspixel}) = \rho \frac{T - 255}{255} \]

**eq(7-13)**

Therefore, combining the effects of illumination, dirt and paint damage, we have three expressions defining the performance of the thresholding algorithm

\[ m(x, y) = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{c(x,y) - T}{\sigma \sqrt{2}} \right) \right) + \rho \frac{255 - T}{255} \]

**eq(7-14)**

and
the results of which will affect the performance of the linking algorithm and ultimately the whole system.

\[
n = \frac{1}{2} \left( 1 + \text{erf} \left( \frac{T - \beta}{\sigma \sqrt{2}} \right) \right) + \rho \frac{T - 255}{255}
\]

eq(7-15)
7.5.2 Module 2: Linker

The performance parameters for the linking module are as shown:

Figure 7-9 Interaction of parameters with the linker module

Since in the linking algorithm, a pixel which is below the threshold can never be linked, the necessary performance transfer function is thus the probability of true region pixels above the threshold not being linked, and spurious pixels from outside the region being linked.

We consider a pixel at $x,y$, whose probability $p(x,y)$ of being in the linked image is equal to the probability that it is above the threshold, $1 - m(x,y)$ (as $m(x,y)$ is the probability that it is below the threshold), multiplied by the probability that at least one of its neighbours is above the threshold and is already linked to the seed pixel. Then:

$$p(x, y) = 1 - m(x,y) \left[ 1 - \prod_{i,j=-1,0,1} 1 - p(x+i, y+j) \right]$$

This is similar to a Markov chain of first order, for which [102]

$$P(\xi_n = j|\xi_0 = i_0, \xi_1 = i_1, \ldots, \xi_{n-1} = i_{n-1}) = P(\xi_n = j|\xi_{n-1} = i_{n-1})$$

eq(7-16)
However as $m(x,y)$ is not constant over the field it is not readily evaluated, and we must approximate its behaviour.

The graph in Figure 7-10 shows $m(x,y)$ in the region where the illumination gradient is close to the threshold, $T$.

![Figure 7-10 Graph showing the probability of a pixel having a grey-level above the threshold, in the region where the illumination gradient intersects the threshold.](image)

If we assume that $g(x,y)$ is approximately linear in this area, we can think of the surface $g(x,y)$ as being a plane which is intersecting the plane $z=T$ along the line $g(x,y)-T=0$. From the graph, where $g(x,y)-T=0$, $m(x,y) = \frac{1}{2}$, and this is where we would expect the edge of the thresholded, but unlinked, region to lie.

For a vertical edge, we can assume that both $m(x,y)$ and $p(x,y)$ are constant in the $y$ direction, such that $m(x,y) = m(x,y+\gamma)$ and $p(x,y) = p(x,y+\gamma)$, where $\gamma$ can be any value. As the region we are trying to link is expected to be roughly circular with a large radius, we shall consider the edges to be locally straight. We thus need only consider what happens in the direction normal to the edge. From equation eq(7-16).
\[p(x, y) = [1 - m(x, y)] \prod_{i, j=-1, 0, 1; i \neq j} 1 - p(x + i, y + j)\]

However if \(p(x, y) = p(x, y+y)\), this reduces to

\[p(x, y) = [1 - m(x, y)]\left[1 - [1 - p(x-1, y)]^3[1 - p(x, y)]^2[1 - p(x+1, y)]^3\right]\]

eq(7-19)

This is a non-linear difference equation which cannot be solved analytically. However it can be evaluated numerically across an edge using local values of \(m(x,y)\) and an initial estimate for \(p(x,y)\), and the assumption that \(p(x,y) \to 1\) as \(x \to \infty\) and \(p(x,y) \to 0\) as \(x \to -\infty\). A solution for \(p(x,y)\) across an edge where \(m(x,y)\) is locally linear and gently sloping is shown below. The dotted line is the corresponding value of \(1-m(x,y)\).

Figure 7-11 Probability function of a pixel being above the threshold, \(1-m(x,y)\) (dotted line) and corresponding probability of being linked, \(p(x,y)\) (continuous line). The pixel position denotes the distance perpendicular to an edge in the illumination profile crossing the threshold at an angle such that eq(7-18) and eq(7-19) hold.

Although the function between \(m(x,y)\) and \(p(x,y)\) is not analytical, a few of its properties are worth noting:
1 - m(x,y) ≥ p(x,y)  

\[ \text{eq}(7-20) \]

As the gradient of \( g(x,y) \) becomes steeper, \( p(x,y) \rightarrow 1 - m(x,y) \). This occurs around 'clean' edges, such as where there is a step in \( m(x,y) \), \( p(x,y) \approx 1 - m(x,y) \).

As the gradient, \( g(x,y) \) becomes shallow, such as where there is a gentle illumination change very close to the threshold, \( p(x,y) \) tends to a step function, such that

\[ p(x,y) \approx 1 - m(x,y) \quad m(x,y) < 0.3 \]  

\[ \text{eq}(7-21) \]

\[ p(x,y) = 0 \quad m(x,y) > 0.3 \]  

\[ \text{eq}(7-22) \]

Similarly, the relationship between the probability, \( n \), of a spurious pixel being above the threshold, and it being linked, \( q(x,y) \) is

\[ q(x,y) = n(x,y) \left[ 1 - \prod_{i,j=-1,0,1 : i \neq j} 1 - q(x+i, y+j) \right] \]  

\[ \text{eq}(7-23) \]

In most of the image, \( q(x,y) \rightarrow 0 \), assuming a seed pixel within the target region is used. Only pixels which border the target region have a significant probability of being linked, and then their effect is small unless large areas of the cap have become obscured or discolored, for example due to large scale paint deterioration or big flakes of concrete landing on the drum.

7.5.3 Module 3: Location of Centre

The algorithm locating the centre of the drum takes the linked image \( l(x,y) \) and finds its centroid position, \( x,y \). The performance transfer function is the relationship between \( p(x,y) \) and \( q(x,y) \) and the error in \( x \) and \( y \), \( \Delta x \) and \( \Delta y \).
The centroid of the linked region $l(x,y)$ is calculated as

$$
\begin{align*}
    x &= \frac{\iint x l(x,y) \, dx \, dy}{\iint l(x,y) \, dx \, dy} \quad \text{(eq 7-24)} \\
    y &= \frac{\iint y l(x,y) \, dx \, dy}{\iint l(x,y) \, dx \, dy} \quad \text{(eq 7-25)}
\end{align*}
$$

where $l(x,y)$ is defined as

$$
\begin{align*}
    l(x,y) &= 1 \quad \text{pixel is part of linked region} \\
    l(x,y) &= 0 \quad \text{pixel is not part of linked region}
\end{align*}
$$

From the analysis of the linking algorithm, the probability of any pixel being part of the linked region is thus $p(x,y)+q(x,y)$. Therefore the expected centroid position calculated by the algorithm is

$$
\begin{align*}
    x_{\text{calc}} &= x_{\text{true}} + \Delta x = \frac{\iint x[p(x, y) + q(x, y)] \, dx \, dy}{\iint p(x, y) + q(x, y) \, dx \, dy} \quad \text{(eq 7-26)} \\
    y_{\text{calc}} &= y_{\text{true}} + \Delta y = \frac{\iint y[p(x, y) + q(x, y)] \, dx \, dy}{\iint p(x, y) + q(x, y) \, dx \, dy} \quad \text{(eq 7-27)}
\end{align*}
$$
7.6 Results

Although a transfer function has been found which will estimate the probable error in drum location given the values of the parameters, a complete distribution for the parameter values is not available. It would require large numbers of video images taken from the SGV in the waste repository to calculate realistic distributions and hence the true error rate for the drum location algorithm. Unfortunately because of commercial confidentiality issues, as well as economic and technical ones, these images are not available. As a demonstration of the analysis, some likely parameter values for this problem have been estimated and used to calculate the performance for a sample of the search space.

7.6.1 Imaging System Model

The imaging system model which was described in section 7.4.3 and used to calculate the illumination profile $ax^2+bx+cy^2+dy+exy+f$, the mean background intensity immediately surrounding the target region, $\beta$ and the noise standard deviation, $\sigma$, as the lighting condition were varied. The results are shown in Table 7-2.

<table>
<thead>
<tr>
<th>condition no</th>
<th>ambient lights</th>
<th>spot light</th>
<th>illumination (lux)</th>
<th>$a$</th>
<th>$b$</th>
<th>$c$</th>
<th>$d$</th>
<th>$e$</th>
<th>$f$</th>
<th>$\beta$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.018</td>
<td>4.3e-4</td>
<td>-1.4e-4</td>
<td>0.042</td>
<td>3.1e-4</td>
<td>1.4e-4</td>
<td>45.4</td>
<td>28</td>
<td>1.06</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0.41</td>
<td>0.017</td>
<td>-6.3e-5</td>
<td>0.147</td>
<td>4.5e-4</td>
<td>-2.8e-5</td>
<td>78</td>
<td>48</td>
<td>1.41</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0.52</td>
<td>0.333</td>
<td>-5.1e-3</td>
<td>-0.209</td>
<td>0.012</td>
<td>6.9e-5</td>
<td>202</td>
<td>70</td>
<td>9.04</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0.018</td>
<td>7.4e-4</td>
<td>-1.3e14</td>
<td>0.032</td>
<td>1.2e-4</td>
<td>-1.8e-4</td>
<td>46</td>
<td>28</td>
<td>1.06</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>0.41</td>
<td>0.02</td>
<td>-1.8e-4</td>
<td>0.149</td>
<td>6.1e-4</td>
<td>-2.4e-5</td>
<td>78.9</td>
<td>47</td>
<td>1.46</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>2</td>
<td>0.52</td>
<td>0.33</td>
<td>-5.1e-3</td>
<td>-0.206</td>
<td>0.012</td>
<td>-1.3e-5</td>
<td>202</td>
<td>71</td>
<td>9.08</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
<td>0.055</td>
<td>-0.036</td>
<td>5.7e-4</td>
<td>0.008</td>
<td>1.0e-3</td>
<td>-1.3e-5</td>
<td>63.2</td>
<td>33</td>
<td>1.62</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>0.51</td>
<td>-0.018</td>
<td>-4.1e-4</td>
<td>0.116</td>
<td>1.4e-3</td>
<td>5.6e-4</td>
<td>95</td>
<td>48</td>
<td>2.03</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>2</td>
<td>0.66</td>
<td>0.288</td>
<td>-5.1e-3</td>
<td>-0.232</td>
<td>0.012</td>
<td>9.3e-4</td>
<td>206</td>
<td>71</td>
<td>9.22</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>0</td>
<td>0.18</td>
<td>-0.035</td>
<td>-6.1e-4</td>
<td>-2.4e-4</td>
<td>1.1e-3</td>
<td>4.2e-4</td>
<td>63.8</td>
<td>33</td>
<td>1.59</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1</td>
<td>0.53</td>
<td>-6.3e-3</td>
<td>-3.5e-4</td>
<td>0.12</td>
<td>1.0e-3</td>
<td>3.7e-4</td>
<td>90.9</td>
<td>50</td>
<td>1.91</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>2</td>
<td>0.72</td>
<td>0.29</td>
<td>-5.0e-3</td>
<td>-0.23</td>
<td>0.012</td>
<td>1.3e-3</td>
<td>206</td>
<td>72</td>
<td>9.18</td>
</tr>
</tbody>
</table>

Table 7-2 The parameters describing the bi-quadratic fit for each of the 12 illumination conditions tested. The number of background lights and spotlights is shown, along with the illumination measured by a light meter during the test.
Each lighting condition was given a number, which will be used to refer to that particular set of conditions. The geometry of the test set up was designed to simulate the conditions in the nuclear drum store, though given limited information from the store itself, for reasons of commercial confidentiality, they are only approximate. The ambient lights column in Table 7-2 refers to the number of strip lights in the lab which were switched on. When no strip lights were on, only a small amount of light entered the lab. The strip lights were then switched on in turn, starting with the light nearest to the test bench. Thus the first light to be switched on had a greater contribution to the overall illumination than did the last. This resulted in the non-linearity in measured illumination levels with respect to the number of ambient lights switched on. Again, the spot lights column refers to the number of spot lights which were switched on. These results also exhibit a non-linearity in the measured illumination level with respect to the number of spot lights, also caused by the two spot lights not contributing equally to the measured illumination. This was caused by the asymmetrical test set-up. In order to capture a picture of the light meter in the image being taken, for recording purposes, it was necessary to place the light meter slightly to one side of the drum cap. Thus the first spotlight to be switched on illuminated the light meter more than the second. Although this experiment was not an ideal model of the actual operating conditions, as it is difficult to relate it back to what happens in the plant, it can be used to test the model of the algorithm.

A small number of images were available from BNFL which had been taken during testing. These had both spotlights on under good flood lighting and are approximately equivalent to condition number 12.

7.6.2 Estimated Performance

As an example, the performance of the system under one of the specified sets of conditions is evaluated below. This is a step by step illustration of the analysis of condition 11, where ambient lighting is good, but one of the spotlights has been switched off to simulate a failure.

The original image and the results from the actual algorithm at each stage are shown on the left below. The corresponding quadratic illumination surface for the given
lighting conditions, and the resulting predicted values for \( m(x,y) \) and \( p(x,y) \) when \( \rho = 0 \) (no dirt on lid) are shown on the right.
Figure 7-13 Original Image

Figure 7-14 Thresholded version of original image, $A_{\text{min}}/A = 95\%$

Figure 7-15 Above image after linking. The non-connected pixels have been removed.

Figure 7-16 Illumination surface

Figure 7-17 Predicted values of $m(x,y)+n(x,y)$ corresponding to the thresholded image. White areas indicate $m(x,y)+n(x,y) = 1$, black areas $m(x,y)+n(x,y) = 0$.

Figure 7-18 Predicted values of $p(x,y)+q(x,y)$, scaled as above.
Actual error (continuous line) and predicted error (broken line) in the drum position with respect to the certainty in apparent drum lid size, \( A_{\text{min}}/A \), is shown in Figure 7-19. As can be seen, on this image, the predicted error is about 40% higher than the actual error. This is most likely due to the illumination model not predicting some of the missing pixels in the top left hand corner of the real linked image. Although they represent a deterioration in the image quality, these missing pixels actually shift the centroid of the region closer to the true position. Hence the actual positioning error is smaller than predicted. This error estimate could probably be improved by using higher order fits to the illumination profile.

![Actual and Predicted Positioning Error](image)

Figure 7-19 Graph of the actual positioning error (continuous line) and predicted positioning error (broken line) in locating the drum cap as a function of the uncertainty in apparent cap size, \( A_{\text{min}}/A \). The bold line at 3mm shows the tolerance of the SGV grabber to mislocation. The effect of other parameters, such as dirt, are fairly small.

As can be seen from the graph, even with a spotlight failure, the positioning error does not exceed the grabber tolerance value of ±3mm, until \( A_{\text{min}}/A \) reaches 85.5%. This would require errors in the SGV positioning system to be in excess of around 35mm. An increase in randomly distributed noise due to dirt or dust does not
significantly alter this until \( \rho \) reaches quite high values. Under testing, the maximum error in the SGV guidance system was 19mm. Although a full distribution of the operating conditions is not available, it would thus seem that performance of the image processing system in locating the drum within tolerance is likely to be dominated by other, non-imaging parameters which we are unable to predict.

7.6.3 Non-Imaging Parameters

As was discussed at the start of this chapter, several of the parameters expected have a severe effect on system performance are very hard to model and predict. These include damage to the electronics and CCD camera due to radiation, multi-path signal effects and signal degradation, which are mostly to do with the hardware and communication system used for the SGV. As the expected error rate due to the image processing system is so small, as large errors in the SGV positioning system, or large amounts of dust are required to generate an error, these non-imaging system parameters will prove dominant in the system performance.

7.7 Summary

This chapter has shown how the vision component of an existing IP system used to locate drums can be characterised. The results produced were compared to the empirical results from the actual algorithm. The prediction was found to give a 40% discrepancy compared to the actual error, under the conditions tested. This prediction could be improved by using a higher order model of the illumination gradient. It showed that the vision system would not fail except in very extreme conditions. However, it is anticipated that many other non-imaging parameters would cause system errors. For this reason, although the methodology has predicted the performance characteristics of the image processing algorithm, it is unable to estimate an overall performance figure for the entire system.
Chapter 8: Conclusions

8.1 Summary

This thesis has described a new technique for the design and performance assessment of image processing systems. It has surveyed the current state of the art in the performance characterisation and benchmarking of IP systems, and shown where it is insufficient in addressing the needs of system developers who have to analyse real-world problems. The thesis then introduced a new methodology to meet this deficiency, which guides system designers and users in formalising the IP problem, and in gathering the appropriate data that describe the operating conditions. It has shown how the overall performance of a system can be described as a Bayesian probability, as a function of operating conditions and system performance, and how the two can be decoupled. The methodology can guide developers in breaking complex systems down into smaller modules. It provides a novel way of analysing these modularised systems, by considering data propagated through the different components as being accompanied by appropriate quality measures. The technique then demonstrates how the interaction of the performance of these modules with each other can be determined, while accounting for the external operating conditions and the internal tuning parameters. In determining these interactions, it then shows how the final performance of a vision system can be predicted.

The methodology was then applied to four different real world IP problems. The first involved using a template matching based tracking algorithm, to track ladles of steel in a steel plant. This was a reasonably simple algorithm, which could be readily modularised. However the operating conditions were complex and difficult to quantify. Consequently, an empirical evaluation of one of the modules was necessary, using proxy measures for the illumination conditions. This gave a measure for the overall performance of the proposed system which, although approximate, was accurate enough to show that the design would not function reliably under the operating conditions. This enabled the system engineers to halt development along these lines at the earliest possible stage.
The second problem that the methodology was used on was intended to supersede the ladle tracking system design in the previous chapter. A different approach to tracking the ladles was used. Rather than tracking them continuously, they were to be identified using special identification markings, by cameras placed at different positions around the plant. This algorithm was more complex, and involved a more in-depth modularisation of the system. However the operating conditions were better constrained and easier to model quantitatively. This enabled an analytical analysis of the system to be carried out, and graphs of predicted error rates as a function of different operating conditions to be calculated. The final distribution of the operating conditions for this problem could not be measured, as video footage of the proposed system’s operating conditions was not available. Estimates of the conditions though showed that error rates of less than 2% could be expected using the current camera layout. Furthermore, because of the detailed nature of the performance data, the operating conditions in the plant (camera geometry, identification marking design etc.) can now be adjusted to substantially improve the performance of the system. A design based on the techniques proposed in the chapter is currently being developed for use in the steel plant.

The third application involved the use of IP techniques to determine the lens aberrations in a transmission electron microscope (TEM). This essentially required the fitting of a synthetic diffractogram pattern to data gathered from a TEM, and again a template matching technique was used as an example of a possible algorithm design. The algorithm itself was simple, and operated within the well-constrained confines of a TEM. The image data was also readily quantifiable, and theories had already being developed describing its expected form. The expected performance of the algorithm was then predicted analytically using quite sophisticated models of the image, and this performance prediction was shown to correspond well with the experimental results.

The fourth system analysed was the drum location algorithm used in the nuclear fuel repository. This was readily modularised and a model predicting its performance was developed. Although the overall performance figure could not be established due to a lack of actual video footage, it was estimated that error rates for the IP system would be very low. Performance would therefore be dominated by some of the many less
predictable factors affecting the system as a whole, such as multi-path radio effects and errors in the positioning system.

Finally, in appendix B, it is demonstrated how the methodology could be applied to a further two IP tasks: ovarian cancer diagnosis and CCTV intruder detection. This shows how wide a variety of problems could be analysed using the approach described in the methodology. The ovarian cancer diagnosis system highlighted one of the limitations of the methodology however, that it cannot be readily applied to algorithms which require training. The methodology only attempts to measure overall performance using two functions, namely operating conditions and algorithm performance. It does not attempt to take into account training data, which is necessary if machine learning, statistical or neural classifiers are used, and where the classification rule is not known at the time of performance evaluation. The methodology is currently being applied to ovarian cancer detection systems by other researchers, who are using classifiers based on pre-determined rules. The intruder detection example showed how a system designed around a series of filters could be modularised. It also demonstrated how a very complex set of outdoor imaging conditions could be modelled using a series of metrics describing the different sizes and velocities of artefacts in the image, and appropriate internal quality propagation parameters could be chosen.

8.2 Conclusions

- This thesis has shown that the methodology has been successful in estimating the performance of the example IP problems. It is applicable to a wide variety of vision tasks and systems, though there may be further limitations in certain cases.

- The methodology can also be used on a variety of levels, depending upon the amount of effort which the user is prepared to put in and the accuracy of the predictions required. The ladle tracking example gave a fairly approximate estimate of the performance, as the effect of several of the parameters were not analysed. However it was sufficient to determine that the system was not viable. Conversely, the lens aberration analysis used more complex modelling, as more precise estimates of the errors were required.
• Parts of the methodology are redundant when analysing certain types of problems. Very simple systems may not benefit from modularisation, if they operate effectively as one module. Also, very constrained operating conditions may make parameter determination seemingly trivial, though it is still a vital part of the performance characterisation process. Under these circumstances, the methodology can be simplified to suit the task at hand. This should make it a useful guide for the performance analysis of most IP tasks.

• There are still likely to be difficulties when using the methodology with data which is very difficult to characterise. Although the ladle tracking example showed that complex data can be approximated using proxy measures, this required writing the template matching module, to establish the empirical relationship between the contrast and the correlation signal depth. This was reasonably quick and simple to perform. However it is anticipated that if an algorithm is effectively a single module, and the data is particularly ill-constrained, the effort required to characterise the performance could increase to the point where it is almost as time-consuming as building and testing the system itself. Under these circumstances, the benefits of using the methodology will be reduced considerably.

• The work has also demonstrated that the methodology is capable of producing meaningful performance surfaces. Coupled with application specific constraints (e.g. camera cost, performance requirements, computational overheads, etc.) the most cost effective, reliable operating point could be derived via constrained optimisation.

8.3 Recommendations for Further Work

There is scope for several lines of research following on from this work. The most immediate of these is to use the methodology to evaluate as many different IP tasks as possible. This will highlight any areas where the methodology has limitations, and may give indications of how it can be improved. The main deficiency that was discovered in the research was the failure to cope with algorithms which involve training. Tackling this could prove difficult. Knowledge of the classification rules are needed to predict performance, as well as information on the clustering of the training
and testing data. However, this often cannot be determined until after the algorithm has been trained and used, by which time the problem has become not one of performance prediction, but testing.

The ultimate aim of this research is to have off-the-shelf algorithms accompanied by comprehensive performance data, which can then be combined by developers to create complete vision systems. The performance data will allow engineers to predict the overall system performance rapidly, without the need to model or test each of the constituent modules. Once this has been achieved, optimisation techniques can then be used to determine the optimum operating point, which can then be used to improve the design. This research has brought that goal a step closer, principally by reducing such complex systems to a series of simple modules, each of which can be characterised independently, and whose performance can be propagated through the system. What is now required is agreement on a system of common input and output performance variables for different types of algorithms or modules. Developers can then describe the performance of their algorithms with respect to these variables. These would then enable modules to be ‘plugged in’ to any system, and the performance readily determined. This is not a simple task. Defining a comprehensive range of performance metrics for any type of algorithm may be possible (an example was given in chapter 3). However, deriving a general, quantitative description of the relevant properties of input image data is very difficult to achieve. The properties which will determine the performance of an IP system are often not only numerous, but also complex to describe. As IP work progresses, this may be possible in certain specialised fields, e.g. cancer diagnosis in medical imaging. Extending this to all aspects of IP will prove a challenging area of research, and one from which the field will benefit greatly.
Appendix A: Error Correcting Codes

This appendix contains a description of the linear and cyclic error correcting codes referred to in chapter 2. It also describes the development of a synchronisation-error-correcting code used for the ladle identification task in chapter 5.

Linear Codes

A linear \((n,k)\) code over \(Z_2\), the finite field of integers modulo 2, is a subspace of \(V_n(Z_2)\), where \(V_n\) is the vector space of \(n\)-tuples with elements from \(Z_2\), spanned by \(k\) basis vectors. Hence all codewords, \(x\), can be written as a linear combination of the \(k\) basis vectors, \(v_1 \ldots v_k\), as shown below:

\[
x = m_1v_1 + \ldots + m_kv_k
\]

Since a different codeword is obtained for each different combination of coefficients, making the message \(m = [m_1m_2\ldots m_k]\) enables our message to be encoded using the above expression.

A code can be generated from any set of \(k\) linearly independent vectors. The task of generating linear codes is choosing the correct set of basis vectors for efficient (high code rate) and robust (maximum distance) encoding and decoding.

Constructing a Linear Code

If the \(k\) basis vectors of our linear code form the rows of a generator matrix \(G\), the notation can be simplified by describing the codeword \(x\), for the message \(m\) as

\[
x = mG
\]

By choosing a matrix \(G\) of the form \([I_k A]\), where \(I_k\) is the identity matrix of dimension \(k\) and \(A\) is matrix \(k \times n-k\) to be determined, the first \(k\) digits of the codeword are the same as the message. The remaining \(n-k\) digits are the check bits. Given any message \(m\), the corresponding codeword can be generated and transmitted.
The received codeword is $x_r = x + e$, where $e$ is an error vector of length $n$ which we assume for the moment has a maximum of one non-zero bit in any position, corresponding to our transmission error. The codeword, $x_r$ is then decoded by using the parity check matrix, $H$. The check matrix, $H$, is an $n-k \times n$ matrix created so that when multiplied by the transpose of any codeword, $x_r$, the transpose of the product, called the syndrome, $s^T$, will be zero.

$$Hx_r^T = s^T = 0$$

It can be shown that [78] given a generator matrix of the form $[I_k \ A]$ the parity check matrix can be defined to be $H = [A^T \ I_{n-k}]$.

Upon decoding, two different results may occur:

If $s^T = 0$, $e = 0$, and we assume no error occurred. The message can be recovered as simply the first $k$ bits of $x_r$.

If $s^T \neq 0$, an error has occurred which may be recoverable.

By calculating $s^T$ and comparing it to the parity check matrix $H$, if only one error has occurred, the column of $H$ which $s^T$ matches corresponds to the position in $x_r$ in which the error has occurred. The original message $x$ can be recovered by reversing the bit in this position and the message, $m$ recovered from the first $k$ bits.

An elegant implementation of this problem was used by Hamming in his $(7, 4)$ code [80]. By a slight rearrangement of the columns of $H$, each column is the binary encoding of its own position.

$$H = \begin{bmatrix} 0001111 \\ 0110011 \\ 1010101 \end{bmatrix}$$
The columns 1, 2 and 4 of \( \mathbf{x} \) now contain the parity bits and 3, 5, 6 and 7 the message. The attraction of this arrangement is that the syndrome \( s^T \) that is generated now indicates the binary number of the position of the error.

In the previous section it was noted that to detect \( t \) errors, the minimum distance between code words had to be at least \( t + 1 \), and to correct \( t \) errors the minimum distance had to be \( 2t + 1 \). Since the codewords belong to a subspace we know that they are closed under addition. If, for example, 1000000 is a basis vector it is also a codeword. Thus \( \mathbf{x} + 1000000 \) is also in the code for any codeword, \( \mathbf{x} \). But the distance between \( \mathbf{x} \) and \( \mathbf{x} + 1000000 \) is only 1 since they differ only in the first bit. This suggests that for a code with a minimum distance of \( d \) all our basis vectors must have weight greater than or equal to \( d \). A linear code which for some \( t \) has all patterns of weight \( t \) or less as coset leaders is called a perfect code. The Hamming code described above is a perfect code. A binary code is optimum if its probability of error is minimised for given \( k \) and \( n \).

Other optimum codes with greater larger values of \( n \) and \( k \) have been discovered. A few of these are listed below [78]:

<table>
<thead>
<tr>
<th>(n, k, d)</th>
<th>(n, k, d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(5,2,3)</td>
<td>(23,12,7)</td>
</tr>
<tr>
<td>(11,4,5)</td>
<td>(23,14,5)</td>
</tr>
<tr>
<td>(15,9,3)</td>
<td>(29,19,1)</td>
</tr>
<tr>
<td>(21,12,5)</td>
<td>(31,20,5)</td>
</tr>
</tbody>
</table>

Table A-1 Length, \( n \), number of information bits, \( k \), and distance, \( d \) of some optimum codes (from [78]).

Cyclic Codes

A code is called cyclic if \( \mathbf{x} = [x_n x_0 x_1 \ldots x_{n-1}] \) is a codeword whenever \( [x_0 x_1 \ldots x_{n-1} x_n] \) is also a codeword. To give us the structure required in these codewords, polynomial descriptions are used. Blocks or vectors with 1's and 0's are used as messages and codewords but they are processed as polynomials. In other words, the four digit message, \([1010]\) would be converted to \( 1 + x^2 \), then encoded into a larger code.
polynomial for processing and then converted back to a vector. All the polynomials in a binary code have coefficients in the Galois field \( \mathbb{Z}_2 \). Other than that, addition, multiplication and division are performed as usual.

**Code Generation**

To encode a message word in a cyclic code, first the corresponding polynomial must be found and then multiplied by a generating polynomial. The generator polynomial is chosen according to [78]:

\[
g(x) \text{ is the generator polynomial for a linear cyclic code of length } n \text{ if and only if } g(x) \text{ divides } 1 + x^n.
\]

Therefore to find a generator polynomial \( g(x) \), \( x^n + 1 \) is factored for whatever length of codeword, \( n \), is required. If \( n \) is the length of the code, \( k = n - \deg(g(x)) \) where \( \deg(g(x)) \) denotes the degree of \( g(x) \). The code polynomial for \( c(x) \) corresponding to a message polynomial \( m(x) \) in this code is then

\[
c(x) = m(x)g(x)
\]

\text{eq}(A-4)

The code generated by \( g(x) \) has a minimum distance \( d = \text{wt}(g(x)) \) and is capable of correcting a maximum of \( t \) errors where, \( 2t + 1 = d \).

**Decoding**

The received code is then decoded as follows:

Compute the syndrome \( s(x) = r(x)(\mod g(x)) \) where \( r(x) \) is the received word. Then for each \( i = 0 \), calculate \( s_i(x) = x^i s(x)(\mod g(x)) \) until a syndrome, \( s_j \) is found for which \( \text{wt}(s_j) \leq t \), where \( t \) is the number of errors to be corrected by the code. Then the most likely error polynomial is \( e(x) = x^{n-j} s_j(x)(\mod (1+x^n)) \).
The existence of, and maximum possible error-correction performance of any
code is bounded by several theoretical limits, the sphere-packing [105], Elias, Plotkin
[78] and other bounds.

Comma-Free and Other Specialised Codes

Error-correcting codes can be also designed to have special characteristics, or be
maximally robust to different types of error. These include tolerance to cluster-errors
[83], where the errors are expected to occur in blocks or clusters, and array codes robust
to column erasures [84], such as might occur during spillage's of steel in this
application. One of the more useful features is the possibility of generating codes that
are comma-free. Comma-freeness refers to the ability of code to be interpreted without
knowledge of where it starts or ends; the codes require no comma's to mark their start
and finish. This may occur for example if a transmitted signal is out of synchronisation
with a receiver, and also if the true location or orientation of a 2D code is not known.
They were originally proposed by Francis Crick [106], considering the possible make up
of a genetic coding for protein synthesis. They have since been developed as a branch of
coding theory [85-89] using different encoding and decoding techniques. One such
technique is based on the use of cyclic codes with a superimposed synchronisation
pattern [78].

Since a BCH code is a cyclic code, it has degree 0 comma-freeness, i.e. any
synchronisation slip would cause it to be read as another code. If we want to introduce a
degree $s$, of comma-freeness, we must generate a synchronisation code, $f(x)$. The
encoding process now works as follows. The codeword, $c(x)$, corresponding to the
message $m(x)$ is generated as $c(x)=g(x)m(x)$. The synchronisation code, $f(x)$, is now
added, under the rules of closed addition (effectively an exclusive-OR operation) and
the code $f(x)+c(x)$ is transmitted.

When the received codeword, $r(x)$, arrives, the known synchronisation code, $f(x)$,
is subtracted from $r(x)$. If there was no synchronisation slippage, and less than $t$ errors,$r(x)-f(x)$ should be a decodable word and can be decoded in the normal way. If there was
synch slippage, there will now be an extra number of errors, between $wt(f(x))$ and
$2\times wt(f(x))$, introduced by the synch code. If $f(x)$ is chosen such that $wt(f(x)) > t$, the
synch slipped code will not be decodable. In this case, the synch code $f(x)$, should be multiplied by $x^s$, where $-T \leq s \leq T$ and $T$ is the maximum expected slippage, before the subtraction and decoding step is repeated. When a decodable codeword is found, the value of $s$ which yielded the codeword will correspond to the degree of synchronisation slippage in the received word, assuming that no more than $t$ errors occurred during transmission and that the polynomials $f(x)x^s$ have a minimum distance, $t$, in the range $-T \leq s \leq T$. This technique and the polynomials given in eq(A-5) and eq(A-6) were used to generate the bullseye codes discussed in chapter 5.

There exists a binary BCH code, with length $n = 32$, information bits, $k = 5$ and error correcting capability, $t = 5$. The generator polynomial can be shown to be [107]:

$$g(x) = x^{26} + x^{25} + x^{22} + x^{18} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x$$

eq(A-5)

Using this with a synchronisation code described by the polynomial

$$g(x) = x^{31} + x^{28} + x^{25} + x^{22} + x^{19} + x^{16} + x^{13} + x^{10} + x^7 + x^4 + x$$

eq(A-6)

enables us to generate a cyclic code capable of decoding 7 bit errors and robust to a degree of synch slippage up to 1 bit position in either direction. This was used to generate a new set of 2-D error-correcting codes which are robust to rotation errors. An example of the `bullseye` code, whose performance is analysed in chapter 5, is shown in Figure A-1.

Figure A-1 Error correcting Bullseye code used in chapter 5. The code information is contained in the teeth. The inclusion of a degree of ‘comma-freeness’ enables the code to be read if it has an unknown rotation.
Appendix B: Further Examples

This appendix contains another two brief examples of how the methodology could be applied to other real-world IP problems. It does not attempt to estimate their performance fully; it just shows how their external parameters can be estimated, the system modularised and the quality measures determined. The first example is taken from the field of medical imaging, and the second from a surveillance system.

B.1 Ovarian Cancer Detection

B.1.1 The Problem

Biopsy samples are taken from tumours in ovaries, which are then placed on microscope slides. These slides are examined by a pathologist who classifies them as either malignant, borderline or benign. The task is to design an image processing system to improve the discrimination of the borderline and benign cases made by the pathologist. Dr. J Barker is currently working on such a system in UCL. The task is essentially one of classification with appropriate choice of feature vectors and preprocessing. It was suggested by Dr. David Lowe of St Bartholomew's Hospital, one of the expert pathologists in the field, that the shape of the epithelium in the sample would be one of the key features in determining which of the classes the sample will fall into. He also suggested that measurements of cell area might be important, as they had been found to be useful in the diagnosis of breast cancer using similar techniques. These are also the features which pathologists tend to concentrate on when making their classification. The image processing system should therefore extract measurements of cell size and epithelium shape which can hopefully be used for classification.
Figure B-1 An example of a specimen taken from an ovarian tumour (50×). The image consists of two regions of interest, the epithelia and the stroma. The epithelium is the dark stained border region. The stroma is the speckled grey area seen here in the top left and bottom right. The uniform grey region is where no tissue is present. Cell nuclei within the epithelium are weakly visible upon greater magnification.

Examples of the different classes of tumour are shown below from [103].

1. *Benign.*
   (25×) No (few) cell-level malignant changes are visible. The epithelium is a smooth, even layer a few cells thick. The epithelium may undulate slightly, but does not appear too complex in structure. Cell adhesion is good, with few cells breaking away from the epithelial surface.

2. *Borderline.*
   (25×) Cell-level malignant changes are apparent. There is (slight) thickening of the epithelium in places, and proliferation of the epithelium produces a complex, convoluted shape reminiscent of a coastline. Cell adhesion may be impaired, with small islands or single cells breaking away from the epithelium. However, there is no invasion of the stroma by the proliferating epithelium.
3. **Malignant.**

(25×) Cell-level malignant changes are apparent. The proliferation of the epithelium is often obvious with large two-dimensional masses of epithelial cells present. Cell adhesion is impaired. Distinguishing malignant from borderline, there is *invasion of the stroma* by the epithelium.

Dr. Barker devised a scheme, which operates as follows: An illumination compensation algorithm divides the grey level of each pixel by the mean of its neighbours’ levels. This compensated image is passed to a thresholding algorithm, the threshold of which, \( T \), is chosen by the user. All regions with areas that are below a certain pre-selected size \( A \), are discarded. This is assumed to have segmented out the epithelium, and is used to generate a mask \( m(x,y) \). The region of the original image corresponding to the masked-out region of epithelium is now passed to another algorithm. This attempts to locate the cell nuclei by scanning for the local minimum within a \( p \times p \) window. The positions of these cell nuclei are then used to form a Voronoi diagram, which gives a rough approximation of the cell sizes by producing the bisects of the lines between a cell nucleus and its neighbours, then intersecting the set of half-spaces which result. An example of the Voronoi cell for a set of cell nuclei is shown in Figure B-2 (from [103]).

![Voronoi diagram generated from cell nuclei](from [103]).

The area of each of these cells is then calculated, and the mean area passed to a classification algorithm.
The segmented epithelium is also passed to a boundary analysis algorithm. This fits a series of quadratics to the epithelium boundary such that the RMS error between the discrete, pixelised boundary and the quadratic fit is less than some pre-determined error, \( e \). An example of a boundary fit is shown in Figure B-3.

![Figure B-3 Quadratic boundary fit to a segmented region of epithelium (100×) (from [103]).](image)

This boundary is then used to generate the following statistics:

<table>
<thead>
<tr>
<th><strong>Length</strong></th>
<th>The total ‘length’ of the boundary (number of pixels)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Curvature</strong></td>
<td>The first and second moments of curvature. The curvature of a given segment is calculated from the length of the second derivative of the quadratic which approximates it.</td>
</tr>
<tr>
<td><strong>Node Count</strong></td>
<td>The number of points where the approximation is split.</td>
</tr>
</tbody>
</table>

**Table B-1 Statistics generated by the boundary detection algorithm (from [103]):**

The length and curvature are then combined to generate the average curvature. These two statistics, average boundary curvature and mean cell size are then passed to a classifier for discrimination between benign and malignant tumours.
B.2 Performance Analysis

B.2.1 Problem Level

At the problem level the main parameters affecting performance are as follows:

<table>
<thead>
<tr>
<th>Image Parameter</th>
<th>Rank severity</th>
<th>Rank occurrence</th>
<th>Effect on Image</th>
<th>Possible Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Illumination gradient</td>
<td>5</td>
<td>1</td>
<td>Change in illumination across image</td>
<td>Grey level variation across image</td>
</tr>
<tr>
<td>Amount of stain used on slide</td>
<td>4</td>
<td>1</td>
<td>Darkness of artefacts on slides</td>
<td>Mean grey level and variation across image, also nuclei and epithelium contrast.</td>
</tr>
<tr>
<td>Cell nuclei contrast</td>
<td>1</td>
<td>1</td>
<td>Ease of visibility of cell nuclei</td>
<td>Grey level difference between nuclei and cytoplasm</td>
</tr>
<tr>
<td>Epithelium contrast</td>
<td>2</td>
<td>1</td>
<td>Ease of visibility of epithelium</td>
<td>Grey level difference between epithelium and stroma</td>
</tr>
<tr>
<td>Dirt on slide</td>
<td>3</td>
<td>5</td>
<td>Dark spots on image</td>
<td>Number, size, contrast</td>
</tr>
<tr>
<td>Epithelium thickness</td>
<td>6</td>
<td>6</td>
<td>Too small number of epithelium cells</td>
<td>Thickness (in cells)</td>
</tr>
</tbody>
</table>

Table B-2 Parameters affecting the performance of the ovarian cancer detection system.

From these, only the effect of epithelium thickness can be neglected from the point of view of severity and probability of occurrence. The performance metric is the receiver operating characteristic (ROC), or the false alarm/misdetect rate for the tumour classifier. The main parameters can all be reasonably well quantified and the problem is quite well constrained. It therefore falls into category two.

B.2.2 System Level

The system modularises into the following components:
Module 1. Illumination compensator - removes slowly varying intensity changes.
Module 2. Epithelium segmentor - thresholds at T, then removes regions with area less than A.
Module 3. Nuclei extractor - finds local minima and labels them as cell nuclei.
Module 4. Voronoi tessellator - generates Voronoi diagram from nuclei locations.
Module 5. Cell size estimator - calculates mean cell size from the Voronoi diagram.
Module 6. Curvature estimator - calculates mean curvature from segmented epithelium image.
Module 7. Classifier - distinguishes between benign and malignant tumours based on epithelium boundary curvature and cell size. A simple linear discriminator was used in the research.

The entire interaction of the information and performance parameters is shown in Figure B-4.

Figure B-4 Diagram showing complete information streams and performance parameters for the ovarian cancer detection system.

Suggestions for the internal quality parameters have been added to the diagram. The probability of mask error, $p(i,j)$, could be the probability of a non-epithelium pixel
classified as epithelium, and vice-versa. Similarly the probability of nuclei error, \( p(n) \) and \( p(m) \) could be the probability of a non-nuclei artefact, such as dirt on the slide, being classified as a nucleus and that of a true nucleus being missed. There is also a mean error on the difference between the reported nucleus position and its true position. These would then lead to differences between the true curvature and the calculated curvature, \( e(Q) \) and the true and calculated cell size, \( e(s) \).

Most of this fairly complex system can be broken down into analysable modules using the methodology. However, there are issues regarding ground truth, for example, determining the true nuclei positions and measuring the errors that are introduced by a hand segmentation. Many of these issues are highly contentious and will not be addressed here. Ground truth issues aside however, all the modules apart from the classifier appear reasonably amenable to performance assessment. Despite the fact that a full performance analysis of each of the modules is not carried out here, the methods used in the other chapters could be adapted to most of this problem.

The classifier however presents a difficult problem for this methodology. The methodology cannot be applied to algorithm stages which involve a training or learning stage. If the classification rule and correlation of the data is already known, then a transfer function between the input data error and the output misclassifications could be determined. In most cases however, it is not, as the classification rules are not predetermined but are generated based on the training data, and the classifier is being used to find the correlation between data and classes which is not already known. Finding a solution to this problem would be a very useful, though difficult, challenge to further work in this field.

**B.3 Intruder Detection**

**B.3.1 The Problem**

The second problem is an example from the field of video surveillance. Cameras are placed at regular (~100 metre) intervals around the perimeter of a prison. The perimeter consists of two fences with a ‘sterile’ zone between them, where there should be no human activity. The cameras cover these sterile zones, and the task of the IP
system is to detect any human motion within the sterile zone. The problem is being investigated by S Young and the Police Scientific Development Branch (PSDB) and is described in [104].

![Image of intruder detection system](image)

**Figure B-5 Typical image from the intruder detection system showing an intruder (centre) entering the sterile zone between the two fences on the left and right sides of the image. [Image courtesy of PSDB]**

### B.3.2 Description of the IP

The region in which the system must detect movement is supposed to be sterile, i.e. there should be no other movement in the region. However in practice, as the perimeter is outdoors, this is generally not the case. The IP system must discriminate human motion from various other non-human movement, such as waving grass, birds, clouds, lighting variations etc.

The system works as follows: An algorithm compensates for camera shake by tracking the apparent movement of certain fixed points in the scene, then shifting the image to (hopefully) eliminate the effect of movement of the camera. An illumination compensation algorithm removes slowly varying illumination changes due to differences in daylight, cloud shadows, etc. A filter then detects small scale temporal variations in illumination which could be human movement. These are then filtered again by two
more algorithms, one filtering out the moving objects based on size (if they are too large or small to be human) and the second based on trajectory (if they are not moving in a locus which could be human movement (e.g. if they are flying or moving too quickly.) All events that have not been filtered out are then signalled as alarms.
B.4 Performance Analysis

B.4.1 Problem Level

After examining images and talking to the developer, Stewart Young, the main parameters affecting the performance of the intruder detection system were estimated as given in Table B-3.

<table>
<thead>
<tr>
<th>Image Parameter</th>
<th>Rank severity</th>
<th>Rank occurrence</th>
<th>Effect on Image</th>
<th>Possible Metrics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera shake</td>
<td>6</td>
<td>1</td>
<td>Whole image translates and rotates</td>
<td>Frequency, amplitude of shake.</td>
</tr>
<tr>
<td>Daylight changes</td>
<td>8</td>
<td>2</td>
<td>Changes in daylight cause slow variations in illumination and shadows</td>
<td>Rate of change of mean grey level.</td>
</tr>
<tr>
<td>Floodlight changes</td>
<td>7</td>
<td>8</td>
<td>Floodlights switching on or off cause rapid variations in illumination and shadows</td>
<td>Rate of change of mean and local grey level.</td>
</tr>
<tr>
<td>Cloud shadows</td>
<td>10</td>
<td>6</td>
<td>Large, sometimes rapidly varying shadows appear in image</td>
<td>Rate of change of mean and local grey level, shadow size.</td>
</tr>
<tr>
<td>Snow in image</td>
<td>5</td>
<td>9</td>
<td>Small flecks appear in image with varying trajectory</td>
<td>Fleck size and velocity. No of flecks.</td>
</tr>
<tr>
<td>Rain in image</td>
<td>12</td>
<td>7</td>
<td>Small flecks appear in image with roughly constant trajectory</td>
<td>Fleck size and velocity. No of flecks.</td>
</tr>
<tr>
<td>Rain/snow on lens</td>
<td>9</td>
<td>10</td>
<td>Sudden changes in image appearance, followed by large parts of the image obscured</td>
<td>Percentage of image obscured.</td>
</tr>
<tr>
<td>Moving vegetation</td>
<td>11</td>
<td>3</td>
<td>Small scale movements in large regions of the image</td>
<td>Frequency and size of moving objects, amount of image moving.</td>
</tr>
<tr>
<td>Birds</td>
<td>1</td>
<td>4</td>
<td>Moving objects in image. Physical size cannot be determined using ground plane map</td>
<td>Projected size in image plane, velocity.</td>
</tr>
<tr>
<td>Land creatures</td>
<td>2</td>
<td>5</td>
<td>Moving objects in image. Physical size can be determined using ground plane map</td>
<td>Projected size in image plane, velocity.</td>
</tr>
<tr>
<td>Target (intruder) size</td>
<td>4</td>
<td>11</td>
<td>Apparent size of target in image varies depending on position in ground plane and viewing angle e.g. intruder may be crawling</td>
<td>Projected size in image plane</td>
</tr>
<tr>
<td>Target (intruder) motion</td>
<td>3</td>
<td>12</td>
<td>Intruder may alter direction and velocity.</td>
<td>Velocity, frequency of changes, length of time velocity is constant</td>
</tr>
</tbody>
</table>

Table B-3 Parameters affecting the performance of the intruder detection system.
There are the following modules:

Module 1. Image Stabiliser - Takes in camera image, performs local inter-frame cross-correlation to determine image movement and shifts image to reduce image shake.

Module 2. Illumination Compensation - Performs a local grey level normalisation on the stabilised image.

Module 3. Temporal Filter - Subtracts consecutive images to detect changes in the image.

Module 4. Spatial Filter - Filters out all moving objects in the differenced image below a pre-determined size. Uses a model of the ground plane to change the size of the filtered objects based on estimated distance from the camera.

Module 5. Trajectory Filter - Measures the trajectory of the spatially filtered regions of interest to generate list of trajectories.

Module 6. Classifier - Classifies trajectories as alarm events or non-alarm events based on length of trajectory.

**Figure B-6 Information stream and suggested performance parameters for the intruder detection system**

The complete interaction of the external parameters and modules is shown in Figure B-6. Suggested quality propagation parameters for each of the stages have also been added. The problem is category 4. It can be broken down and analysed, but a lot of the effects, such as vegetation motion, would be easier to analyse empirically, by taking sample images and measuring the response of the temporal filter. Given a reasonable
estimate of the probability distribution of the parameters should enable the performance to be predicted.

**B.5 Summary**

This appendix has shown how the methodology described in chapter 3 can be applied to a further two problems from other areas of image processing, namely medical imaging and video surveillance. Although it did not go into detail in analysing the final performance of the systems described, it has shown how the principals of parameter determination, modularisation and quality propagation can be applied more generally to other IP tasks. Is has also highlighted a limitation of the methodology: it is unable to predict the performance of classifiers if the classification rule is not known.
Bibliography


