Statistical Modelling of Colour Data and Model Selection for Region Tracking

Daniel Alexander

Submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy from University College London.

1997

Department of Computer Science UCL, Gower Street, London. WC1E 6BT.
Abstract

This thesis is concerned with the question of how best to model naturally occurring distributions of digital colour camera data so that objects may be characterised in terms of such data and thus tracked through sequences of images. By use of a physical model of the processes by which colour data is obtained, several statistical models, based on directional statistics, are proposed as alternatives to the standard multivariate Gaussian distribution. Various types of algorithm are discussed that can use these statistical colour models for tracking regions of interest through sequences of images. One such region tracking technique, the active region model or statistical snake, is investigated in detail and some improvements to the original model are suggested.

In order to decide which of a set of candidate statistical models and which of a set of region tracking algorithms is the best, a methodology for measuring the performance of these algorithms in terms of pixel classification is developed. This methodology is used to optimise the internal parameter settings of several region tracking algorithms and to compare their overall performance. For the connected, compact regions of interest in the test set, active region models are shown to outperform simpler thresholding and region growing algorithms in terms of pixel misclassifications. Some of the suggested alterations to the existing active region model implementations are shown to increase performance. However, it is also found that the introduction of some desirable properties, such as stability and convergence, can increase susceptibility to local minima and reduce overall classification performance.

The set of candidate SCMs are compared over two data sets, representative of two separate applications, by comparing the performance of region tracking algorithms using each candidate model. It is shown that in controlled environments directional models are superior to Gaussian models. However, for imagery obtained in less constrained daylight environments, generalisation of the directional models is found to be poor and the Gaussian model proves to be preferable.
Acknowledgements

For his invaluable input to the work documented in this thesis and his unceasing encouragement and enthusiasm, I would like to thank my supervisor, Bernard Buxton. I would also like to thank Ben Southall, Ismail Ismail and Chris Harris for their thoughts on various parts of various drafts of this thesis. My second supervisor, Simon Arridge, from UCL, Jon Crowcroft, also from UCL, Will Hallinan, DERA Chertsey, and Jon Savage, also at DERA Chertsey, also deserve thanks for general advice and input during the extent of this project.

Special thanks go to Roheena Anand for her continual support and tolerance, particularly over the last few months, and to Lee Wright for his gracious acceptance of the usurpation of his printer.

Further acknowledgements go to Jim Ivins, currently working in the Artificial Intelligence and Visual Research Unit at the University of Sheffield, for the kind donation of his statistical snake code. Similarly, to John Kent, at the University of Leeds, and Christopher Bingham, University of Chicago, for their code used to perform maximum likelihood estimation for the parameters of the Bingham distribution.

This project was funded by an EPSRC CASE studentship supported by DERA Chertsey.
B.2.1 Black Pixels ................................................................. 218
B.2.2 Look-up Tables ............................................................ 219
B.2.3 Computation Times ...................................................... 219
B.3 CLUSTERING PROCEDURE .............................................. 220
  B.3.1 Histogram Storage ...................................................... 221
  B.3.2 Histogram Resolution .................................................. 221
  B.3.3 Pruning ...................................................................... 222
B.4 REGION GROWING .......................................................... 223
B.5 COMPUTING GOODNESS INTEGRALS ......................... 223
  B.5.1 Line Integrals .............................................................. 224
  B.5.2 Extended Line Integral ............................................... 225
  B.5.3 Surface Integrals .......................................................... 226
  B.5.4 Extended Surface Integral ......................................... 228

APPENDIX C RECEIVER OPERATING CHARACTERISTIC ANALYSIS ............................................. 232
  C.1 THE ROC CURVE .......................................................... 232
  C.2 SYSTEM MODEL .............................................................. 234
  C.3 PERFORMANCE MEASURES FROM ROC CURVES .......... 237
  C.4 OPERATING POINT SELECTION ........................................ 238
  C.5 PROBLEMS .................................................................. 238

APPENDIX D FURTHER RESULTS ................................................................. 240
  D.1 CLOTHES TRAINING SET, INTENSITY MODEL COMPARISON ........................................ 240
  D.2 CLOTHES TRAINING SET, CHROMATICITY MODEL COMPARISON ..................................... 241
  D.3 TRACKS TRAINING SET, INTENSITY MODEL COMPARISON ................................................ 243
  D.4 TRACKS TRAINING SET, CHROMATICITY MODEL COMPARISON ........................................ 244
List of Figures

Figure 1 Geometric picture of a typical viewing scenario. 22

Figure 2 Models for the shape of the distributions of SPDs of light reflected from a convex body under a single illumination. 23

Figure 3 Two images containing regions of interest that need to be characterised by a SCM. The seed region is outlined in each image. 48

Figure 4 RGB cubes showing the occurrence histograms of data in (top) the seed regions and (bottom) the regions of interest as a whole throughout the sequences of Figure 3. 49

Figure 5 Gaussian models constructed from the sample distributions shown in Figure 4. 53

Figure 6 Directional chromaticity, uniform intensity SCMs constructed from the sample distributions shown in Figure 4. 55

Figure 7 Sample (top) and parent (bottom) chromaticity distributions, from the images of Figure 3, shown in the unit plane. 55

Figure 8 Gaussian chromaticity models in the unit plane. 56

Figure 9 Demonstrates how tighter models of directional distributions can be constructed by choosing a plane perpendicular to the mean of the distribution. 57

Figure 10 Gaussian chromaticity models in planes oriented with the mean of the sample distributions. Shown in the unit plane. 58

Figure 11 ZYCUI models constructed from the sample distributions shown in Figure 4. 59

Figure 12 Angular normal chromaticity models depicted in the unit plane. 61

Figure 13 Bingham chromaticity models depicted in the unit plane. 63

Figure 14 Robust Bingham models depicted in the unit plane. 64

Figure 15 Disparity of directions of lines from the origin caused by uncertainty in the absolute position of a data point. 64

Figure 16 Chromaticity models constructed using Healey’s approach, depicted in the unit plane. 65

Figure 17 BCGI models (top) and ZYCGI models (bottom) constructed from the sample distributions shown in Figure 4. 68

Figure 18. Rough surface illuminated and viewed from fixed directions 69

Figure 19 Sample (top) and parent (bottom) distributions of intensities from the objects in Figure 3 with uniform, Gaussian and planar intensity models. As usual, the pictures on the left correspond to the green shorts and those on the right to the track. 71

Figure 20 Grey level likelihood images generated from a Gaussian SCM, with the tolerance parameter, k, set to a moderate value of 3.0, applied to the images in Figure 3. 105

Figure 21 Region segmentations generated by thresholding the two grey scale likelihood images in Figure 20 at a value of 120. 105

Figure 22 The effects of binary mathematical morphology on the region segmentations shown in Figure 21. A morphological closure is performed followed by an opening using a circular structuring element of diameter 9 pixels. 105
Figure 23 Regions extracted using the region growing algorithm of this section with a Gaussian statistical colour model with k set to 3.0.

Figure 24 Effects of mathematical morphology on the regions extracted by the region growing algorithm, shown in Figure 23. A morphological closure is performed followed by an opening using a circular structuring element of diameter 9 pixels.

Figure 25 The four subsequent region segmentations for the track sequence following those in Figure 23 (top – no morphology) and Figure 24 (bottom – morphological post-processing).

Figure 26 Typical region segmentations obtained from an ARM applied to the images of Figure 3.

Figure 27 Evolution of two ARMs one (top) using Ivins and Porrill's collision resolution strategy and one (bottom) using element deletion.

Figure 28 Various configurations of a variational ARM having reached the desired image boundary.

Figure 29 Typical graphs of the average force magnitude versus iteration number. $\alpha$ is fixed at 1.0 and $\beta$ and $\rho$ (the image energy coefficient) have the values given in the legend.

Figure 30 AFM versus iteration number for CID and DAG ARMs with identical parameter settings applied to the first image of the track sequence.

Figure 31 Three elements of a contour, $\mathbf{u}$, on an image are shown as is the set of candidate new positions for the middle element. The triangles that would be included (or excluded) by one of these moves are shown.

Figure 32 Typical steady state of a SLG ARM showing convergence of some elements and some simple oscillations and cycles.

Figure 33 Graphs of number of moving elements against iteration number showing the four different steady states.

Figure 34 Shows the situation on the subsequent iteration to that shown in Figure 31, when the middle element has been moved up one pixel, but the other two remained fixed.

Figure 35 Contours extracted by DAG (top) and MLG (bottom) ARMs from the images of Figure 3.

Figure 36 Contours extracted by the MLG ARM with k increased to 3.5.

Figure 37 Ground truth data for the two images shown in Figure 3.

Figure 38 Illustration of the common algorithm model used in ROC analysis.

Figure 39 The first image of each sequence from the clothes data set. The seed regions are superimposed on each region of interest.

Figure 40 The first image of each sequence from the tracks data set. The seed region is superimposed on each.

Figure 41 ROC curves from hand region segmentations.

Figure 42 Plot of performance of the thresholded likelihood image algorithm over the tracks training set as the diameter of the structuring element is varied.

Figure 43 Plot of performance of the region growing algorithm over the tracks training set as the diameter of the structuring element is varied.
Figure 44 Plot of $z(A)$ for the DAG ARM applied to the tracks test set with a wide range of parameter settings. 154

Figure 45 Performance measures obtained from each ARM over the tracks training set. 156

Figure 46 Plot of the ARM performances over the tracks test set. 157

Figure 47 ROC curves obtained for each ARM over the tracks test set. 158

Figure 48 ROC curves from the best algorithm of each type generated over the tracks test set using a Gaussian SCM. 159

Figure 49 ROC curves generated by the ARM over the clothes test set using various intensity models 161

Figure 50 ROC curves obtained from the ARM over the clothes test set using various chromaticity models. 162

Figure 51 Plot of ARM performances using various statistical colour models over the clothes test set. 163

Figure 52 ROC curves generated by the ARM over the tracks test set using various intensity models. 164

Figure 53 ROC curves obtained from the ARM over the tracks test set using various chromaticity models. 165

Figure 54 Plot of ARM performances using various statistical colour models over the tracks test set. 166

Figure 55 ROC curves obtained from thresholded likelihood images, with no morphological post-processing, over the clothes test set. 167

Figure 56 Plot of thresholded likelihood image performance over clothes test set for various SCMs. 168

Figure 57 ROC curves obtained from thresholded likelihood images, with no morphological post-processing, over the tracks test set. 168

Figure 58 Plot of thresholded likelihood image performance over tracks test set for various SCMs. 169

Figure 59 An example illustrating how low measures of performance arise for the OIP ARM (shown in purple), which is compared to the CID ARM (shown in blue). 171

Figure 60 Evolution of a DAG ARM, over the first four frames of a sequence – from left to right, top to bottom, with (in red) and without (in blue) the termination condition applied. 173

Figure 61 An example of an MLG ARM (shown in purple) getting stuck at a local energy minimum position, which is passed over by the DAG ARM (shown in blue). 174

Figure 62 An unstable final configuration caused by the fixed topology of the ARM (shown in blue). 176

Figure 63 Likelihood images obtained, from the shorts image of Figure 3, using uniform (left) and Gaussian (right) intensity models and a directional (Bingham) chromaticity model. 178

Figure 64 Likelihood images obtained, from the shorts image of Figure 3, using Gaussian chromaticity model in a fixed plane (left) and an oriented plane (right). 178
Figure 65 Likelihood images obtained, from the shorts image of Figure 3, using Healey’s model (left) and an angular normal model (right).

Figure 66 Likelihood images obtained, from the shorts image of Figure 3, using a trivariate Gaussian model (left) and a Zhu and Yuille SCM (right).

Figure 67 Likelihood images obtained, from the tracks image of Figure 3, using uniform (left) and planar (right) intensity models and a directional (Bingham) chromaticity model.

Figure 68 Likelihood images obtained, from the tracks image of Figure 3, using Healey’s chromaticity model (left) and a linearly down-weighted Bingham model (right), both with planar surface intensity models.

Figure 69 Likelihood images obtained, from the tracks image of Figure 3, using a trivariate Gaussian SCM and Zhu and Yuille’s chromaticity model with a planar surface intensity model.

Figure 70 A multicoloured object under a single illumination (left) on a cluttered background.

Figure 71 Occurrence histograms of colour data showing the sample (left) and parent (right) distributions from the object of interest in Figure 70.

Figure 72 Single mode Gaussian (left) and BCUI (right) models constructed from the seed region shown in Figure 70.

Figure 74 Multimode statistical colour models for the region of interest in the image shown in Figure 70.

Figure 75 Sample (left) and parent (right) chromaticity distributions from the region of interest in Figure 70.

Figure 76 Single mode (left) and multi-mode (right) Bingham chromaticity models for the region of interest in Figure 70.

Figure 77 ROC curves for single mode and multi-mode Gaussian models.

Figure 78 ROC curves for single mode and multi-mode BCUI models.

Figure 79 ROC curves for single mode and multi-mode BCGI models.

Figure 80 The atomic sub-pixel regions over which expressions for the surface integral are required.

Figure 81 Typical triangle over which the surface integral might be required shown on a 2D image grid.

Figure 82 The sixteen ways in which one of the edges of the triangle can intersect a consistent square.

Figure 83 Examples of ROC curves.

Figure 84 Depicts the distributions of measurements made by a system on a set of test cases and how the application of a decision threshold gives rise to each type of decision outcome.

Figure 85 An example showing how a non-convex ROC curve can arise.
List of Tables

Table 1 Summary of directional chromaticity models. * indicates that a chromaticity model has not been used previously and so is an original contribution of this thesis. 50

Table 2 Summary of intensity models. * indicates that an intensity model has not been used previously and so is an original contribution of this thesis. 51

Table 3 Summary description of ARM implementations used in this work. * indicates that the ARM is new and so is an original contribution of this thesis. 103

Table 4 Lists the end points of the lines joining to the previous and next elements of the contour, along which the goodness is averaged to ensure single step consistency of the image energy change approximation. 123

Table 5 Calculation of operating points. 148

Table 6 Operating points (values of k) used for each ARM using a Gaussian SCM over the tracks data set. 154

Table 7 Optimal parameter settings for each ARM found on the tracks training set. 156

Table 8 Performance measures obtained for each ARM over the tracks test set. 156

Table 9 Performance measures for each region tracking algorithm generated over the tracks test set using a Gaussian SCM. 159

Table 10 ARM optimal parameter settings and performance measures obtained for each intensity model over the clothes training and test sets 161

Table 11 Optimal parameter settings, maximum performance over clothes training set, and final performance over clothes test set for the ARM using various chromaticity models. 162

Table 12 ARM optimal parameter settings and performance measures obtained for each intensity model over the tracks training and test sets. 164

Table 13 Optimal parameter settings, maximum performance over tracks training set, and final performance over tracks test set for the ARM using various chromaticity models. 165

Table 14 Performance of the thresholded likelihood image algorithm over the clothes test set using various SCMs. 166

Table 15 Performance of the thresholded likelihood image algorithm over the tracks test set using various SCMs. 167

Table 16 Values of z(A) found for each model. 212

Table 17 Processor times for likelihood image construction using various SCMs. All times are quoted in seconds. 220
Chapter 1 Introduction

The work reported in this thesis is an investigation into the statistical modelling of naturally arising distributions of digital colour data. The purpose of the investigation is to determine how best to characterise objects in terms of the colour image data that corresponds to them. Such a model can be useful for a number of automatic visual tasks such as the identification and location of objects in a scene; extraction of the image region that corresponds to a particular object; and tracking of such regions of interest through sequences of images. The application motivating this investigation is tracking and, in particular, tracking country roads and lanes to assist the navigation of cross-country autonomous land vehicles.

1.1 Useful Sources of Information

There are several sources of information that might be useful to a system that performs region tracking tasks of this type. Grey-level pixel intensity data can be used to provide some degree of discrimination between different regions in an image and many image processing techniques, such as those discussed later in Chapter 4, often use grey-level data directly. However, when only the grey-level intensity is available, it can often be hard to distinguish separate objects from one another. The use of edge information, typically derived from an image by application of a differential or morphological filter, can greatly assist the task of extracting distinct regions particularly when some knowledge of the shape of the region of interest is held. Active contours, also discussed in Chapter 4, have proved to be an effective way of exploiting grey level data in this way and have become a popular tool for tracking.

Multiband imagery arises when a scene is viewed simultaneously by a number of different registered sensors. Colour images typically have three bands corresponding to three sensors with different response functions – one with its peak in the red part of the visible spectrum, one in the green part and one in the blue. Other sensors may have response peaks elsewhere in the electromagnetic spectrum, e.g., infrared cameras, radar or X-ray machines, and so can provide alternative or extra bands. Each separate band yields a simple intensity image like those discussed in the previous paragraph.

The extra dimensionality in data from multiband images allows a much greater degree of discrimination between different regions in an image than can be provided by single band intensity imagery. For this reason, together with the gradual increase in available computer power making computations involving large amounts of data more feasible, multiband imagery is beginning to be preferred for many applications. In particular, colour image data has proved useful for tracking roads and lanes in the past. Crisman, [Cri92], used colour image data to characterise and track paths, along which an autonomous vehicle was required to navigate,
through sequences of images. Her approach, which is an extension of the colour component of
the road tracking system presented by Thorpe, et al, [THK88], is further discussed in Chapter 5.
A more general review of the use of colour data in computer vision related applications is given
in Chapter 2.

Other information sources that can be useful for tracking include range data, which can
be obtained directly from range scanners or can be inferred from camera data either by using a
stereo pair of cameras or by analysing discrepancies between consecutive frames of image
sequences. In the DROID system, developed by Harris, et al, [Har92], range data, generated by
associating corner features between consecutive frames of an image sequence and analysing the
motion from frame to frame, is used to generate a topological map of the environment. This
map can be used to find roads, or drivable regions, by looking for horizontal planar regions in
that map; it is particularly useful, however, for obstacle detection. The motion of corresponding
points between consecutive frames of image sequences can also be used directly to differentiate
between image regions corresponding to different objects. Smith, [Smi95], exploits this
information, by grouping together regions containing points with similar motion, to find and
track moving vehicles in road scenes. Smith’s system is shown to work well on a range of
imagery and it has also been used, in conjunction with a fixed calibrated camera, to track cars
on a motorway and calculate their speed. Stereo matching algorithms provide range information
by analysing the disparity of associated features in the pair of images and so can be used in a
similar way, as in the navigation system for planetary rovers developed by Matthies, [Mat92].

Another rich source of information is image texture. Texture measurements are
typically made by convolving several filters with raw image data to produce a number of texture
images, which together can be regarded as a multi-band image. This multi-band data can be
dealt with in a similar way to colour imagery, for example, by using statistical methods as in
this work. A popular set of filters for differentiating between different image textures, suggested
by Turner, [Tur86], is derived from the family of Gabor functions. In earlier work due to Laws,
[Law79], an optimal set of filters for discriminating between a number of real-world textures
were suggested which turned out to be approximations to filters constructed from 2D Gabor
functions. Ivins and Porrill, [IP95], use a multivariate Gaussian statistical model to characterise
the distribution of Laws texture measurements arising in an image region corresponding to an
object of interest. Using this model, their algorithm, described later in Chapter 4, is shown to
separate two differently textured regions in a single image with reasonable accuracy. Weldon
and Higgins, [WH96], also use statistical methods to analyse texture measurements, employing
a Rician model of the distribution of Gabor filter responses corresponding to a single textured
region. Again, limited results are presented in which a single image is separated into regions of
different texture. Another popular method for modelling texture information is by use of a
different type of statistical method, the Markov random field. Grunes and Sherlock, [GS90], use
a texture model of this type for identifying drivable regions, such as roads, in images. In a single example, their algorithm shows limited success in classifying each pixel as road, grass, foliage or brick. Campbell, et al, [CMT97] use a neural network model of combined colour and texture data for similar purposes. Their results are more comprehensive, showing good rates of pixel classification, and are discussed in the next chapter.

Despite a common lack of comprehensive empirical testing and evaluation, all the work discussed above suggests that tasks such as tracking can be achieved with some success by properly exploiting any of these sources of information. A higher level of performance and robustness, however, would almost certainly be achieved by combining analyses of all the available information. Crowley and Berard, [CB97], have developed such a system for tracking faces through sequences of images. Colour information, motion detection and frame to frame correlation are all used to pinpoint the location of the face; the different visual processes are supervised by a blackboard architecture. The resulting system is reported to be much less likely to lose track of the object of interest than a system that exploits just one of these complementary sources of information. The road following system of Thorpe, et al, [THK88], exploits both colour and texture information, together with a simple model of the shape of the corresponding image region, to find the road. 3D data from a laser range finder is also used to detect obstacles. Although the system is quite slow, Crisman quotes 6 seconds per image on a sun4, the tracker is reported to be very reliable. There are many issues of concern when constructing systems that fuse information from different sources and Abidi and Gonzales, [AG92], provide an extensive general discussion of these issues. One clear guideline is that to maximise the performance of any system that fuses information from several sources, each contributory information source must be well understood so that it can be fully exploited. To this end, the work in this thesis concentrates solely on colour data and investigates how best to use it for tasks such as region tracking.

Potential applications for robust, reliable tracking systems are wide and varied. As mentioned, tracking of roads or paths through sequences of images is an important step toward autonomous guidance of vehicles or robots. Face tracking systems, such as that of Crowley and Berard, can be useful for video conferencing as the amount of data that has to be transmitted can be reduced if the region of interest in the image is known. Algorithms for tracking lips and other facial features, see for example the work of Terzopoulos and Szeliski, [TS92], could be a crucial component in a system for automatic lip-reading or facial expression interpretation. Similar algorithms for tracking hands and other body parts can be used in gesture recognition systems, as in the work of Heap, [Hea95], which might be used in future user interfaces. Systems for tracking people have also been developed, for example by Baumberg and Hogg, [BH94]. Such systems could be useful for surveillance, or applications like the Smart Kiosk of Rehg, et al, [RLW97], which detects and tracks shoppers in its vicinity, attracts their attention and delivers...
sales pitches. Both people tracking and gesture recognition systems have been employed in interactive virtual environments such as the KIDSROOM, developed by Bobick, et al, [BDI96, IDB97].

Tracking systems also have agricultural and biological applications. Systems capable of tracking stationary crops from a moving vehicle, such as the system described by Reynard, et al, [RWB96], are useful for autonomous agricultural vehicles, both for navigation along rows of crops and to pinpoint individual plants to be watered or fed. Marchant and Onyango, [MO95], developed a system for tracking pigs in scenes, and a similar system has been used, by McFarlane and Tillet, [MT97], for tracking fish. Such systems provide useful monitoring of livestock, which can be used to judge feeding requirements and the timing of grading and harvesting. While working towards the development of a robot livestock herder, Sumpter, et al, [SBT97], used a tracking system to generate data used to analyse the behaviour of herds of animals. Leymarie and Levine, [LL93], developed a system to track cells through sequences of microscopic images in an attempt to automate analyses of cell motion, which is of great importance in some areas of biological research.

Sequences of images through which a region must be tracked are not necessarily temporal and a common medical application of tracking systems is for volume rendering. Given a series of slice images through the brain, for example, a region corresponding to a particular part of the brain might be tracked through the series of slices to yield the 3D shape of that region. An example of this type of application is the work of Ivins and Porrill, [IP94a].

1.2 Thesis Plan

The problem motivating the work reported in this thesis is that of tracking regions of interest through sequences of images. The intended contribution to that problem is to learn how best to exploit one particular source of information – colour image data – in a system that performs tracking tasks. Attention is therefore focussed on the development of a tracking system based solely on colour data, although future inclusion of other types of data is borne in mind. During the course of this investigation, three main questions arise:

- How is an object best modelled in terms of the colour data that corresponds to it?
- Once an object has been modelled in terms of that data, how is that model best exploited in order to track the object through a sequence of images?
- How can “best” be defined?

These questions are addressed in the remainder of the thesis, in roughly the order given above.

Chapter 2 contains background on the use of colour in machine vision. In order to model distributions of colour data effectively, it is necessary to understand the physical processes that give rise to these distributions. Some existing models for these physical processes
Chapter 1 Introduction

are presented and discussed. Given a distribution of colour data that corresponds to an object of interest, there are several modelling techniques available for characterising the object in terms of that data. Chapter 2 goes on to review the use of colour data for common machine vision tasks with emphasis on the modelling techniques used in those applications.

The modelling technique investigated in this thesis is the use of parametric statistical models and in Chapter 3 a number of statistical models for colour data are presented. A simple model of the reflection processes that give rise to colour data is adopted and used to infer the shape of the statistical models. This model implies that the data can be divided into two independent components – a directional chromaticity component and an intensity component corresponding to the magnitude of the combined sensor responses. Directional models for colour data, based on a similar physical model, have been used in the past and previous attempts at modelling distributions of colour data in this way are reviewed. Using established directional statistical theory, however, some alternative, less ad-hoc, directional chromaticity models are proposed. Although statistical models of the distribution of intensity data comprising regions of interest have been investigated to some extent in the past, to the author’s knowledge these have not been incorporated into statistical models for colour data. Existing models for intensity data are reviewed and a novel model for roughly planar surfaces is proposed. The chapter concludes by beginning to address the third question above with some discussion of how the most appropriate model for a particular distribution of data might be identified.

The first of the three questions at the start of this section, that of how best to characterise an object in terms of colour data, is addressed in Chapter 2 and Chapter 3. Chapter 4 and Chapter 5 are concerned mainly with the second – how to exploit such models in order to track objects in scenes. Chapter 4 provides background and review of techniques available for tracking regions of interest through sequences of images and in Chapter 5, the algorithms that are used in this work are detailed. The review in Chapter 4 shows a wide diversity of different approaches to region tracking but also highlights the lack of empirical comparison between the different techniques. This deficiency makes selection of the most appropriate technique somewhat difficult, but, as a starting point, three different algorithms are used. The first algorithm is a simple thresholding technique, the second is a region grower and the third is based on the active region model proposed by Ivins and Porrill, [IP94, Ivi96] – the “statistical snake”. Some problems with the original implementation of the active region model arose during the course of the work and these problems and suggested solutions are the main subject of Chapter 5. Several alternative implementations of the active region model are developed in order to overcome these problems.

The question of which statistical colour model, or which region tracking technique is best cannot be answered in general because different models and techniques will be more
appropriate in different situations. Within a particular application domain in which an algorithm exploits a statistical model to perform some task, however, the best statistical model might reasonably be defined to be the one that maximises the performance of the algorithm. This is how the third question, how to define "best", posed at the beginning of this section is answered here. The best statistical colour model is identified by applying a region tracking algorithm to data representative of certain applications, using a set of candidate models, and choosing the model that produces the highest performance of the algorithm. In Chapter 6, the problem of how to measure the performance of such algorithms is addressed. Existing methods for obtaining performance measures of similar algorithms are reviewed and, by considering the objectives of both the algorithm and the performance characterisation itself, a suitable performance measure, based on receiver operating characteristic analysis, is identified. The experimental protocol, by which the statistical colour models and different region tracking algorithms are to be compared, is also outlined in Chapter 6.

The statistical colour models developed in Chapter 3 are compared over two sets of data representing two separate applications. The first set is comprised of sequences of images obtained under conditions that approximate the assumptions inherent in the simple reflection model used to develop the directional statistical colour models. However, an off-the-shelf camera and illumination source, and real-world objects of interest (items of clothing), are used throughout. The second set consists of sequences of images containing roads and tracks taken in daylight scenes, again using an off-the-shelf imaging device. Experiments and results are described and presented in Chapter 7. From these results comparisons of the performance of various statistical colour models, the three different types of region tracking algorithm and various alternative implementations of the active region model are derived.

Conclusions are drawn in Chapter 8 and Chapter 9 contains some discussion of the achievements of the work presented in this thesis and some possible directions for the future.

1.3 Contributions and Accomplishments

This section summarises the original contributions of the work reported in this thesis.

• Analysis and development of statistical models for distributions of colour camera data.
  • Analytic comparison of existing methods for representation of directional chromaticity data.
  • Use of established directional models, in particular the Bingham distribution, for modelling distributions of chromaticity data.
  • Development of a new, non-Gaussian model for the distribution of intensity data arising from planar objects.
Chapter 1 Introduction

- Combination of chromaticity and intensity models to produce full RGB data models.
- Development of active region models with consistent enough behaviour to enable their performance to be characterised reliably.
  - Identification and amelioration of a significant mode of failure of existing active region model algorithms – the tangling effect.
  - Application of greedy energy minimisation techniques to active region models in order to increase stability and convergence properties.
- Framework for performance characterisation of region tracking algorithms over sets of data representative of particular applications.
  - Comparison of the performance of active region models with simpler thresholding and region growing techniques.
  - Comparison of the performance of different implementations of the active region model.
  - System-level comparison of the performance of the statistical colour models used to drive the region tracking algorithms.
Chapter 2 Colour

The extra information contained in colour images compared to single band intensity images potentially provides much greater degrees of discrimination between different image regions. Accordingly, the use of colour data in systems performing machine vision tasks is becoming increasingly popular. However, the extra dimensionality brings added complexity and before colour data can be exploited properly, some investigation of its particular properties is required. In order to give some insight into the shape and structure of distributions of colour data, in this chapter the physical processes that give rise to and affect these distributions are examined and commonly used models of these processes are reviewed. Representation of colour data is then discussed, followed by a review of the uses to which colour has been put within the field of machine vision. The emphasis of the review is on the variety of techniques by which distributions of colour data can be modelled to characterise objects of interest in scenes and the chapter concludes with some discussion of these options and their appropriateness for the applications of interest here.

Luong, [Luo93], also provides a review of colour machine vision. He reviews representation of colour data and colour constancy more thoroughly and places less emphasis on modelling aspects. Further reviews of the representation of colour data can be found in the books by Wyszecki and Styles, [WS67], and Foley, et al, [FVF90].

2.1 Physical Models of Digital Colour Data

In order to exploit complex multi-dimensional data, such as colour data, to its full potential, the data and the processes that give rise to that data must be well understood. To this end, some models of the physical processes that combine to give rise to the digital colour data found in colour images are discussed in this section.

2.1.1 The Dichromatic Reflection Model

The dichromatic reflection model (DRM) was introduced by Shafer, [Sha85], to describe the way that light interacts with surfaces of non-homogeneous dielectric materials and it is now widely used in machine vision applications when analysing colour data. According to the model, light from a single illuminant reflected by a surface has two components - surface reflected light and body reflected light.

Surface reflected light is reflected at the interface of the surface with the atmosphere and is the component that dominates in highlights or specularities in a scene. The direction of the reflected light ray depends only on the direction of the incident ray, see Figure 1, and the surface normal, N, at the point of impact, which bisects the angle between the directions of the
incident and reflected rays. The direction of a ray of surface reflected light is labelled as the “specular direction” in Figure 1. Surface reflected light is assumed to have constant spectral power density (SPD), up to a scalar factor depending on the geometry of the scene. A common restricted version of the dichromatic reflection model – the neutral interface reflection (NIR) model, which has been shown to be applicable for a wide range of materials, [TW89, LBS90, Tom91], assumes that the normalised SPD of surface reflected light is the same as that of the illuminating light.

![Specular Direction](image)

**Figure 1 Geometric picture of a typical viewing scenario.**

Body reflected light is due to subsurface reflection where light enters the body of the material and gets scattered and partially absorbed by subsurface pigment before eventually being reflected back out of the surface. When the light ray re-emerges from the surface, its direction is much less dependent on the direction of incidence and is often assumed to be independent. This component, therefore, unlike surface reflected light, does not tend to form sharp intensity peaks or highlights. The intensity of the reflected light is, however, strongly dependent on the intensity of the incident illumination, which itself can be strongly dependent on the orientation of the surface with respect to the light source. The SPD of body reflected light is assumed to be, again up to a scalar factor depending on scene geometry, the product of the illuminant SPD and the reflectance SPD of the reflecting surface. From a surface patch with fixed reflectance SPD, \( s(\lambda) \), lying in a scene illuminated by a single illuminant with SPD \( i(\lambda) \), the SPD of the reflected light, or scene radiance, \( L \), assuming the DRM and NIR, is given by

\[
L(\lambda, g) = m_s(\lambda)i(\lambda) + m_b(\lambda)s(\lambda)i(\lambda),
\]

where \( \lambda \) parametrises the wavelength of electromagnetic radiation; \( m_s(\lambda) \) and \( m_b(\lambda) \) are the non-negative scalar coefficients of the surface and body reflection components, respectively, which depend on the scene geometry, \( g \), which incorporates not only \( \theta \) in Figure 1, but also the angles made by the viewing direction with the incident and surface normal directions. \( L \) is therefore constrained to lie on the positive subset of the plane spanned by \( i(\lambda) \) and \( s(\lambda)i(\lambda) \) in the infinite dimensional space of spectral power densities (SPDs).

Assumptions about the distribution of surface patch orientations and incident light directions can further constrain the range of the reflected light SPD. Adopting a DRM, Klinker,
et al, [Kli93, KSK90], consider the case of smooth, single coloured surfaces under a single, fixed, collimated light source. They argue that under these conditions, the range of orientations of surface patches from which a non-negligible amount of surface reflected light is received at a fixed sensor position is highly restricted, giving rise to sharp highlights. The value of $m_b(g)$, and hence the intensity of the body reflected light component, for surface patches corresponding to highlights is roughly constant. The SPD of light reflected from non-highlight regions is constrained to move on a line passing through the origin in a direction given by $s(\lambda)i(\lambda)$. The value of $m_s(g)$, which varies much more quickly than $m_b(g)$, can take a range of values over a highlight region of the image and thus the SPD of light from a highlight region is constrained to move on a line starting at some point on the non-highlight line, heading in a direction parallel to $i(\lambda)$. The overall distribution of reflected light SPDs over the surface then assumes a skewed-T shape, [Kli93, KSK90].

Bajcsy, et al, [BLJ96], generalise this model for rougher surfaces or slightly non-collimated light sources by allowing $m_b(g)$ to vary over the range of $g$ for which $m_s(g)$ is non-negligible. For surfaces like this, the highlight line of the skewed-T shape distribution widens to produce a shape more like a skewed-P. Figure 2(a) depicts the skewed-T shape of the distribution of reflected light SPDs from a smooth surface, predicted by Klinker, et al, and Figure 2(b) shows the slightly more general skewed-P shaped distribution obtained for rougher surfaces.

![Figure 2 Models for the shape of the distributions of SPDs of light reflected from a convex body under a single illumination.](image)

If the surface shape is more complex (i.e., non-convex) and the light source is proximal, although there is still just one body reflection line, there may be multiple highlight lines or clusters.

Often, as argued for example by Matas, et al, [MMK94], and Zhu and Yuille, [ZY96], because of the sparse nature of data with appreciable surface reflection component, it is reasonable to consider only the body reflection component so that:

$$L(\lambda, g) = m_b(g)s(\lambda)i(\lambda).$$

With this very simple, unichromatic, reflection model, the reflected light SPD is constrained to move only on the positive half of a single line passing through the origin.
2.1.2 Sensor model

A light sensor, or camera, samples the SPD of the light entering the camera, or image irradiance, to produce its response. This process is well modelled by a spectral integration:

\[ x = \int \tau(\lambda)L(\lambda)d\lambda, \]  

where \( x \) is the sensor (greylevel) response and \( \tau(\lambda) \) is the sensor response function. A multispectral sensor has a number, \( n \) say, of linearly independent channels, each of which has a separate response function, \( \tau_i(\lambda) \), where \( i=1\ldots n \), giving rise to a separate response

\[ x_i = \int \tau_i(\lambda)L(\lambda)d\lambda. \]  

For colour sensors, \( n \) is usually 3, with the three sensor response functions having peaks in the red, green and blue parts of the visible spectrum. This gives rise to three separate measurements, loosely corresponding to the amounts of red, green and blue light contained in the input signal.

A substitution into (4) can be made from equation (1) to give

\[ x_i = m_s(g)\int \tau_i(\lambda)L(\lambda)d\lambda + m_b(g)\int \tau_i(\lambda)s(\lambda)L(\lambda)d\lambda. \]  

The scalar geometric factors \( m_s(g) \) and \( m_b(g) \) are assumed to be independent of wavelength, \( \lambda \), and therefore are the same for each channel. The separate grey level response values, \( x_i \), are grouped into a response vector, \( x \), at each location in the image grid. It is easily verified, see [Sha82], that equation (4) describes a linear transformation between the infinite dimensional SPD space and the \( n \)-dimensional response space. This linearity ensures that, assuming the DRM, the distributions of data found in the response space, like distributions of reflected light SPDs, are also planar and are parametrised by response vectors corresponding to pure body and surface reflected light. For this to be true, it must also be assumed that the camera itself is linear in terms of the mapping it makes from the incoming light intensity to the image intensity, which has been shown empirically, by Beyer, [Bey92], to be a reasonable approximation for CCD cameras. From (5) it can be seen that as the scene geometry varies, the response vector, \( x \), is constrained to move in the positive quadrant of a plane that passes through the origin and is parametrised by the vectors

\[ \lambda^b = \begin{pmatrix} \int \tau_i(\lambda)s(\lambda)L(\lambda)d\lambda \\ \int \tau_i(\lambda)L(\lambda)d\lambda \end{pmatrix} \quad \text{and} \quad \lambda^b = \begin{pmatrix} \int \tau_i(\lambda)s(\lambda)L(\lambda)d\lambda \\ \int \tau_i(\lambda)L(\lambda)d\lambda \end{pmatrix}. \]  

Thus, from (5) and (6),

\[ x = m_s(g)x^s + m_b(g)x^b. \]  

24
Chapter 2 Colour

The sensor response is therefore also constrained to lie in the positive subset of a plane. When there is no surface reflection component, as in the unichromatic model of equation (2), $\mathbf{x}$ is constrained to just the line passing through the origin in the direction of $\mathbf{x}^b$. Although the actual colour of body reflected light is not fixed as its intensity may vary arbitrarily, the direction of $\mathbf{x}^b$ is fixed. This direction depends only on shape of the surface reflectance and illumination SPDs, and so is fixed for a surface with constant surface reflectance under a fixed illumination. Moreover, responses that lie on the same line, if the effects of sensor noise are ignored for now, can only have arisen from surface patches with the same chromatic properties (i.e., a metameric surface reflectance SPD). Lines radiating from the origin of the sensor response space may therefore be considered to be chromaticity equivalence classes and the chromatic properties of a surface under a fixed illumination be characterised by a unit vector $\hat{\mathbf{x}}$ in the direction of $\mathbf{x}$.

When there is a surface reflection component, the response from a single surface patch is no longer restricted to a line. However, as discussed in the previous section, assumptions about the imaging conditions can further constrain the range over which the sensor response can vary. In such situations, again by linearity of the transformation made by the sensor, the distributions of colour sensor data corresponding to a single surface take on the skewed-T or skewed-P shapes shown in Figure 2. Examples of distributions of colour data taken from real images that exhibit these shapes can be found in [KH93] and [BLL96].

2.1.3 Modelling $m_b$ and $m_s$

Lambert's law states that the intensity of the light incident on a point of a surface is related to the intensity of the illumination by the cosine of the angle of incidence of the illumination on the surface, $\theta$ in Figure 1. Using this simple geometric statement, a common model, used for example by Novak and Shafer, [NS94], for the body reflection coefficient, $m_b$ of equation (1), is given by

$$m_b = C \cos(\theta), \quad (8)$$

where $C$ is a constant. This model makes the additional assumption that the intensity of the scene radiance is directly proportional to that of the light incident on the surface and so does not depend on viewing direction. If the shape of a surface is known then restrictions on the distribution of $\theta$ can be inferred and, assuming the model for $m_b$ given by (8), this can allow further restrictions on the expected distribution of colour data to be placed.

Although appealing because of its simplicity, this model for $m_b$ can be inaccurate, particularly when illumination and/or viewing directions are close to grazing, because of surface roughness causing local vignetting. Vignetting is the masking effect caused by surface points being shadowed by other points of the same surface and it affects both the intensity of the incident illumination at each point and the amount of reflected light that reaches the sensor.
Koenderink and Van Doorn, [KV96] discuss these effects in detail. Wolff, [Wol96], contests the simple model given by (8) even for smooth objects for which vignetting effects are minimal. He shows from first physical principles that the inherent assumption in (8) that the intensity of body reflected light is independent of viewing direction does not hold in general. An alternative, but more complex, model is derived which is shown to correspond more closely to observed data. The difference in the models however occurs mainly when the angle of incidence with respect to the viewing direction is large and Wolff suggests that when this angle is less than about 50°, (8) is a reasonably accurate model.

Various models for the surface reflection coefficient, $m_s$, also exist in the literature. A commonly used model in computer graphics, which produces realistic looking highlights but is not based on any real physical model, is that of Phong, see [FVF90]. A significantly more complex model, derived from basic physical principles and a surface model incorporating roughness and local vignetting, is proposed by Torrance and Sparrow, [TS67]. They show that their model predicts the observed phenomenon of off-specular peaks in scene radiance when viewing and illumination angles are close to grazing.

2.1.4 Applicability of Reflection Models

The DRM is designed for inhomogeneous dielectric materials. Homogeneous materials, like metals and crystals, do not exhibit body reflection; surface reflection dominates for these materials. This surface reflection may be diffuse if the surface is rough and may also be tinged with the intrinsic colour of the material. Although the NIR assumption may be violated for these materials, as long as the shape of the SPD of the reflected light does not change with the viewing and illumination geometries, the chromatic properties are constant and so a unichromatic reflection model still applies. A model, such as that expressed by equation (2) can be used when there is a single illuminant, as the distribution of sensor responses is one-dimensional. Healey, [Hea88], shows empirically that a unichromatic reflection model fits real data obtained from a number of both coloured, e.g., copper and gold, and white, e.g., aluminium and silver, metals.

Other materials can exhibit body reflection but still tinge surface reflected light. Again, the dichromatic model still applies but the NIR assumption does not and equation (2) is still valid in the absence of a significant surface reflection component. Translucent and/or luminous materials are not modelled well by the dichromatic model, however, as there are additional components to the scene radiance. Klinker, [Kli93], provides some further discussion of the range of different types of material in terms of reflection properties.

Tominaga and Wandell, [TW89, Tom91], and Lee, et al, [LBS90], performed empirical studies to show that the DRM, with NIR, provides a good approximation to the reflective properties of a range of materials. Both studies however found seemingly inhomogeneous
Chapter 2 Colour

materials for which the shape of the SPD of surface reflected light varied considerably over a range of illumination and viewing geometries so violating the DRM. Tominaga found that silk and satin materials did not fit the model well, although Lee found it reasonable for other types of cloth. Both found some paper surfaces not to be modelled well; but, in contrast to earlier findings of Lee, at al, Tominaga found ceramic surfaces were modelled well. Even when the surface reflection component varies with the illumination and viewing geometry, the body reflection component may still be modelled well by the unichromatic reflection model of equation (2) in the absence of a surface reflection component. In [Hea88], Healey performs further experiments to verify the integrity of the linearity of the body reflection component of light reflected from inhomogeneous dielectrics, which is assumed in the dichromatic reflection model. Good agreement with the model was reported over a number of colour chips from the Munsell book of colour, [Mun76].

2.1.5 Multiple Illuminants and Daylight

The general dichromatic reflection model and some special cases, like the NIR assumption or the unichromatic reflection model of (2), make no assumption about the type of illumination in the scene and are equally valid for a light source of any shape or dimension. As discussed above however, the more specific models of the shape of distributions of colour data, like the skewed-T models shown in Figure 2, assume a collimated light source. Although the skewed-P model accommodates a slightly wider range of light sources, allowing for some extendedness of the source itself, it becomes less valid for very extended sources. Even in its most general form, however, the dichromatic reflection model does assume that the illumination chromaticity is constant. When there are multiple sources of illumination with different chromaticity, the situation becomes more complex.

Consider the case when there are two distinct light sources with different chromaticity. At any point on the surface of an object of constant surface reflectance in the scene, the incident light is a mixture of the light from each source. The amount of light incident on the surface from each source at any point depends on the surface normal of the surface at that point and may also be affected by shadowing. The product of the surface reflectance with the light mixture gives the chromaticity of the body reflection component at that point, and thus the camera response to purely body reflected light is given by

\[ x^b = \int \frac{r(\lambda)}{\lambda} y_1 i_1(\lambda) + y_2 i_2(\lambda) s(\lambda) d\lambda = y_1 \int \frac{r(\lambda)}{\lambda} i_1(\lambda) s(\lambda) d\lambda + y_2 \int \frac{r(\lambda)}{\lambda} i_2(\lambda) s(\lambda) d\lambda \]  

(9)

where \( y_1 \) and \( y_2 \) are the weights of the two illuminants, \( i_1 \) and \( i_2 \), at that point on the surface. As discussed in section 2.1.2, the colour of the surface under this illumination can be characterised by a unit vector in the response space. As the weights of the two illuminations vary, the unit
vector moves linearly between two extreme points corresponding to the object under each of the illuminations alone. By similar arguments, when there are several illumination sources, if the unichromatic reflection model of equation (2) is used, the unit vector corresponding to the line joining the camera response to the origin lies within the convex hull of the extreme points corresponding to each illumination alone. When the surface reflection component is non-negligible, the situation becomes highly complex, as any mixture of the illuminant SPDs can be found in highlight regions making the overall range of sensor responses to the surface potentially very wide. Further discussion of this type of situation can also be found in [Kli93, BLL96, and FB93].

Daylight illumination is of particular importance for the road tracking application under investigation here. In general, there are two component sources of illumination, the sky and the sun, that have different chromaticity. The sky can be modelled as a hemispherical light source of uniform chromaticity and intensity. The sun may be viewed as a point light source at infinity so that its light is collimated. The scene radiance at a point on a surface under these two illuminations combined is the sum of the light reflected from each. From the collimated source the distribution follows the skewed-P shape shown in Figure 2. From the hemispherical source, it is often assumed that the incident illumination at each point on a surface is the same, leading to the familiar idea of an ambient illumination component. In [Sha85], Shafer suggested the incorporation of an ambient term into the dichromatic reflection model to deal with just this situation:

$$x = x^A + m_s(g)x^s + m_b(g)x^b$$

(10)

where the new term $x^A$ is the (constant) sensor response to reflected light from the sky alone.

Koenderink and van Doorn, [KV96], however, showed that this ambient model is unrealistic because of global vignetting. Points on surfaces illuminated by a hemispherical light source, such as the sky, in fact receive different amounts of illumination because, for most points, other points on the surface will obscure light from some parts of the hemisphere. They illustrate this point with the example of a drilled hole in some surface. As the hole becomes deeper, the portion of the hemispherical light source that is directly visible from the bottom of the hole becomes smaller and smaller and thus the amount of illumination received by points at the bottom of the hole decreases.

This means that the scene radiance from a surface of constant reflectance, rather than being constant, as implied by the ambient model, can have variable intensity. Furthermore, most points on the surface will receive light from their specular direction and so exhibit a reasonable amount of surface reflection as well as body reflection. Some points however may have light from the specular direction obscured; the corresponding scene radiance then lacks the surface reflection component and may have different chromaticity. The situation can be even further
Chapter 2 Colour

compounded if mutual illumination is considered, i.e., light incident on one part of the surface that has already been reflected from another. Bajcsy, et al, [BLL96], consider mutual illumination to some extent, but it is not covered here. It is merely noted here that models of the processes that contribute to characterise distributions of colour data can become arbitrarily complex as more contributory effects are incorporated.

2.1.6 Bi-directional Reflection Distribution Functions

For characterisation of the reflection properties of specific materials, the bi-directional reflection distribution function (BRDF), originally due to Edwards, [Edw63], is a useful tool. The BRDF describes the intensity ratio of the light incident on and reflected from a surface patch of a particular material at each combination of viewing and illumination angles. For a particular surface, points on the BRDF can be found by measuring the sensor response corresponding to the surface at fixed illumination and viewing angles. To approximate the true BRDF, the function is sampled at regular intervals in this way and can be interpolated to obtain a continuous approximation. For colour imagery, three intensity ratios are measured at each point.

If the BRDF of a material is known, the models discussed above are not necessary as the distribution of sensor responses can be calculated directly from the BRDF. Although these intensity ratios are relatively insensitive to changes in illuminant SPD, as shown independently by Funt and Finlayson, [FF95], and Nayar and Bolle, [NB96], the BRDF will be affected to some extent by such changes. More information is incorporated into the bi-directional spectral reflection distribution function (BSRDF), which includes the wavelength of light as a parameter. Given the SPD of the illumination and the illumination and viewing directions, the BSRDF returns the ratio of the incident and reflected SPDs. Generation of a BSRDF for a real-world surface could be an extremely arduous task however and so in practical situations, only BRDFs are used. Koenderink, et al, [KVS96], proposed a set of hemispherical functions that can be used as a basis to represent BRDFs so that the BRDFs of natural surfaces might be more compactly represented. Using this basis, a database of BRDFs for common real world materials is currently being constructed by Dana, et al, [DVN97].

2.1.7 Linear Models and Colour Constancy

A major issue in the use of colour in machine vision systems is the problem of colour constancy. When the spectral content of the illumination on a particular surface changes, so does the SPD of reflected light and hence the distribution of sensor data corresponding to that surface. The goal of a colour constancy algorithm is to deduce the intrinsic colours, i.e., surface reflectance SPDs, of objects in a scene so that they can be recognised under any illumination. Generally, it is only possible to solve the colour constancy problem at regions in an image where there is negligible surface reflected light or where the image has been pre-processed to
remove the surface reflection component using an algorithm such as those proposed in [BLL96, KSK90]. The problem in its most general form may be formulated directly from (6), where, given $x^b$, the problem is to estimate $s(\lambda)$ and $i(\lambda)$. A common assumption in tackling this problem, originally used by Maloney and Wandell, [MW86], is that naturally occurring surface reflectance SPDs and illuminant SPDs can be represented closely by a linear combination of a finite number of basis SPDs, so that

$$i(\lambda) = \sum_{j=1}^{p} a_j i_j(\lambda) \quad (11)$$

and

$$s(\lambda) = \sum_{k=1}^{q} b_k s_k(\lambda). \quad (12)$$

The $a_j$'s and $b_k$'s are then the weights of each basis surface reflectance and illumination SPD respectively, $p$ is the number of basis SPDs required to represent the set of surface reflectances of interest and $q$ is the number required to represent the number of illuminant SPDs. This assumption has been experimentally justified for naturally occurring surface spectral reflectances by Cohen, [Coh64], who performed a principal component analysis on the spectral reflectance SPDs of 150 colour chips randomly selected from the colour chips in the Munsell Book of Color, [Mun76]. He found the first three principal components to cover about 99% of the spectral distribution.

Similar, and more recent, analyses can be found in [Dan92, Mal86, and PHJ89], the most extensive being due to Maloney, [Mal86], who analysed 462 Munsell colour chips and 337 natural reflectance SPDs. Maloney found that five to seven basis SPDs were required to represent samples accurately, but three or four provide sufficient accuracy when the colour of the light is measured by an imaging system such as the human eye or a CCD camera.

Similar analyses on naturally occurring sources of illumination have shown that the assumption is similarly justified. Judd, et al, [JMW64] analysed the SPD of 622 phases of daylight, taken from England and the USA, and performed a principal components analysis showing that the first three principal vectors cover almost 100% of the spectral distribution. Dixon, [Dix78], performed a similar analysis, with comparable results, for Australian daylight. Marimont and Wandell, [MW92], analysed a wider range of illumination source SPDs in the same way and still found that three basis functions were sufficient to represent the whole set.

From the linear models, of (11) and (12), and equation (6), the sensor response to body reflected light can be written.
Chapter 2 Colour

\[ x_b = \sum_{j=1}^{p} \sum_{k=1}^{q} a_j b_k \begin{pmatrix} \int r_i(\lambda) s_j(\lambda) \lambda k(\lambda) d\lambda \\ \int r_i(\lambda) s_j(\lambda) \lambda k(\lambda) d\lambda \end{pmatrix} \] (13)

and only the \( a_j \)'s and \( b_k \)'s need to be estimated to achieve colour constancy. (13) can be solved if enough objects are viewed under sufficient variety of different illuminations. For example, suppose that a sensor provides 3 colour channels, and that 3 basis functions are used to represent both the illuminant and surface reflectance SPDs. If two objects are viewed under two different illuminations, then there are six surface reflectance weightings (\( b_k \)'s) and six illumination weightings (\( a_j \)'s) that must be estimated. A separate instance of equation (13) arises from each object under each view and each instance provides three measurements. Thus there are 12 unknowns and 12 identities so the problem is solvable, at least in theory. This is exactly the approach taken by Tsukada and Ohta, [TO90], and similar approaches with varying numbers of colour channels, basis functions, visible objects and illuminations are discussed by Maloney and Wandell, [MW86], and D'Zmura and Iverson, [DI93, DI93a, DI94].

Other approaches attempt to estimate the illumination SPD directly, either from highlights assuming NIR, e.g., [Lee86, DL86], or by use of a white reference plate in the image whose measured colour is assumed to correspond to the illumination colour, as in the work of Bajcsy, et al, [BLL96]. Once the illumination is known, the surface reflectance can be estimated at every point in the image by solving equation (13) directly provided there are sufficient colour channels (i.e., at least as many as there are surface reflectance basis functions). Probably the most successful approach, originally proposed by Forsyth, [For90], and further developed by Finlayson, et al, [Fin96, BFF96], stores a set of plausible illuminants, which gives rise to a set of plausible surface reflectances at each point of the image. Illuminants are cleverly represented as diagonal transformations of the colour space, representing the transformation that maps the distribution of colour data observed to that under a canonical illuminant. An illuminant is plausible if its transformation maps observed data to a distribution that is physically plausible under the canonical illuminant.

Even the most successful colour constancy algorithms, however, tend to be heavily reliant on restrictive models of reflection and corresponding assumptions upon which they are developed and have only been shown to work in controlled environments. Algorithms that can reliably provide colour constancy in unconstrained environments, such as those illuminated by daylight, are not currently available in the literature.
Chapter 2 Colour

2.2 Colour Spaces and Features

There are many different ways used to represent colour data in the literature. For surveys of
commonly used representations, see the books by Foley, et al, [FVF90], and Wyszecki and
Styles, [WS67]. The simplest representation is the RGB colour cube, which is the representation
in which raw colour data is usually supplied by a colour sensor, see equation (4). Many other
commonly used representations are simply linear transformations of this space. The YMC
space, for example, which is commonly used by printing devices, is defined by yellow, Y,
magenta, M, and cyan, C, axes, with Y=G+R, M=R+B and C=B+G. Ohta, et al, [OKS80],
compared the use of several colour spaces for segmentation tasks. By using the Karhunen-Loeve
transformation to find a set of orthogonal colour features that provide the largest variance over
regions that require segmentation into homogeneous sub-regions, they tried to find a set of
features that provide maximal discriminatory power. Over the eight images in their data set, the
features they found were, (R+G+B)/3, R-B, and (2G-R-B)/2, which again define a space that is
a linear transformation of the RGB space. Segmentation results obtained using these colour
features as opposed to those defining other spaces were deemed, by inspection, to be better than
those obtained using other features. The results of Pietikainen, et al, [PNM96], substantiate this
claim to some extent by finding better recognition rates when using Ohta, et al’s, colour features
rather than RGB and normalised RGB features in some simple colour recognition experiments.
This space is similar to another commonly used linear transformation of the RGB space, the
opponent colour space, defined by (R+G+B)/3, R-G, (2B-R-G)/2. It is so called because its axes
range from black to white, red to green and blue to yellow, respectively, all of which are
opponent colours in human perception. Another common linear transformation of the RGB
space is the YIQ space, used for colour television signals, see [FVF90, Luo93].

Some representations are designed more for human interaction than for use in automatic
systems and tend to be non-linear transformations of the raw RGB data. Two such
representations that are commonly used in graphics packages and word processors are the Hue,
Saturation, Value (HSV) space, and the Hue, Lightness, Saturation (HLS) space. Both are based
on the artist’s model of tint, shade and tone and are designed to allow intuitive navigation
around the set of available colours. Other spaces, such as the CIE (Commision Internationale de
l’Eclairage) LUV space, have been defined in such a way as to approximate human colour
perceptual differences. In such spaces, the Euclidean distance between two points in the space,
at least over small distances, reflects the difference in the colours defined by those points as
perceived by a human observer. Although some attempts to use such spaces in automated visual
tasks have been made, e.g., [GLO94], where statistical models of database objects are
constructed in the HSI space; and [CM97], in which the LUV space is used for image
segmentation, this is not the use for which they are intended. For automatic tasks there is no
obvious benefit in using these representations. In fact, because of the highly non-linear nature of
the transformation from raw RGB sensor data to these perceptual spaces, these representations
may be a significant hindrance to the automatic interpretation of colour data. This is particularly
true when trying to model distributions of data, as non-linear transformations tend to complicate
the shape of the distributions.

Many colour representations contain an axis that represents the intensity of the sensed
light and the independence of this component and the chromatic component of colour data has
been widely noted. Furthermore, the reflection models discussed in the previous section suggest
that the intensity may vary arbitrarily over a region corresponding to a single object of one
colour. For this reason when using colour to distinguish between image regions corresponding
to separate objects in a scene it is common to ignore the intensity component and just use the
other two components that characterise the chromaticity. Accordingly, normalised colour is
often used in vision applications. Two chromaticity values are found by normalising the RGB
vector by the intensity, defined as R+G+B, to obtain the L₁ normalised colour co-ordinates

\[
(r, g, b) = \frac{1}{R + G + B} (R, G, B),
\]

which contain only two degrees of freedom since b = 1 - r - g. The normalised colour space, defined
by the set of all such points, is thus the unit plane in the RGB colour cube. A similar
representation is the 1931 CIE chromaticity plane, [WS67], in which chromaticity is represented
by points, \((\frac{R}{B}, \frac{G}{B})\), which is the intersection of the line joining a data point to the origin
with the plane B = 1. Use of either of these representations is well justified by the theory in
section 2.1, which suggests that lines passing through the origin of the RGB space represent
colours with fixed chromatic properties. The normalised colour space is isomorphic to the space
of lines passing through the origin of the RGB cube, as a unique point on the unit plane
represents every such line and vice versa. The chromatic plane is a less neat representation of
chromaticity as there is no point corresponding to the chromaticity of points for which B = 0.

Healey, [Hea92], noted that more uniform differentiation between chromatic properties
may be achieved by using the L₂ norm, to give

\[
(r, g, b) = \frac{1}{\sqrt{R^2 + G^2 + B^2}} (R, G, B),
\]
rather than the L₁ norm used in (14). As discussed later in Chapter 3, this is particularly
significant when dealing with distributions of data.

Bajcsy, et al, [BLL96], use a colour space based on the dichromatic reflection model. A
white reference patch in the scene is used to find the colour of the illumination and a vector in
this direction, \(\gamma_0\), describes one axis of the space. Two orthogonal vectors, \(\gamma_1\) and \(\gamma_2\) are found

33
and used as the other axes. A normalised space similar to the chromatic plane is obtained by using \( \left( \frac{\gamma_1}{\gamma_0}, \frac{\gamma_2}{\gamma_0}, 1 \right) \), which, assuming NIR, effectively eliminates the surface reflection component from the data.

### 2.3 The Use of Colour in Machine Vision

In this section, a review of the use of colour data in machine vision tasks is provided. Colour is a feature that provides discrimination between image regions corresponding to different real world objects and this discriminatory power has been exploited for a number of different machine vision problems. Usually this involves the construction of a model of image regions in terms of the colour data comprising it and the focus of the review is on the variety of modelling techniques available.

#### 2.3.1 Segmentation

A fundamental task in machine vision is image segmentation. The segmentation problem consists of dividing an image into separate regions corresponding to different objects in a scene. Haralick and Shapiro, [HS85], and Pal and Pal, [PP93], have reviewed established techniques for image segmentation, but in this section a review of the use of colour for segmentation is provided.

A simple way to use colour for image segmentation is to perform a cluster analysis on the colour data in an image. Data from within distinct regions is assumed to be proximal in the feature (colour) space, often drawn from a compact statistical model like a normal distribution, while data from other image regions occupy different areas of the space. The resultant separation of the colour space is projected back to the image to obtain segmentation. Bezdek, et al, [BHC93], review image segmentation techniques for magnetic resonance images, which, like colour images, are multiband, and discuss a number of techniques based on clustering procedures. Early attempts at this approach, such as the method proposed by Coleman and Andrews, [CA79], often use the k-means algorithm, [DH73], to separate the image data into a number (k) of clusters with specified shape. The number of clusters, or distinct image regions, must be specified in order to apply this algorithm although a number of techniques for automatically selecting this number have been suggested for use in image segmentation algorithms, [CA79, ZM90]. Furthermore, a parametric statistical model for the colour data comprising an image region corresponding to each distinct object must also be pre-specified. Other clustering algorithms, such as those of Koontz, et al, [KNF76] and Kittler, [Kit76], do not require either the number of distinct regions or the shape of the clusters of data to be pre-specified. Khotanzad and Bouarfa, [KB90], used the algorithm of Koontz, et al, to segment both grey-level and colour images and reported good results, although output of the algorithm was
shown for only two images and no attempt to quantify the precision of the results was made. Raw segmentations produced in this way often appear noisy and contain misclassified pixels well away from the true image boundaries. Post-processing, for example by morphological operations, or by analysing connected components can ameliorate results to some extent, see [KB90] for example. Matas, et al, [MMK95], extended the work of Khotanzad and Bouarfa by incorporating spatial information into the algorithm. Efficacy of the resulting algorithm is demonstrated on some synthetic data but again no quantitative results or comparisons with other methods are shown. As might be expected, however, segmentations produced on examples of real images appear less noisy when spatial information is incorporated in this way. Further review of these techniques can be found in Appendix A.

Ohlander, et al, [OPR78], use a region splitting approach to segment colour images into object and background. Image regions are iteratively subdivided while a suitably distinctive peak can be found in any one of a number of histograms of the values of different colour features taken over that region. The region is thresholded to extract pixels with values corresponding to the histogram peak and then connected components are extracted to define new smaller regions. This is the approach to image segmentation used by Ohta, et al, [OKS80], to find the set of colour features that provide maximal discriminatory power for colour images in general, which was mentioned in section 2.2. The region model in this segmentation algorithm is simply that the colour vectors are within some distance, defined by the thresholds used, of each other and that the region is connected. The segmentation technique based on this model therefore will not reliably separate the image into regions corresponding to distinct objects. Highlights will tend to be separated from surrounding regions and sudden intensity changes, caused for example by shadowing, within a region corresponding to a single object may cause the region to be split.

In [Sha85], Shafer suggested a method by which a colour image might be separated into surface and body reflection component images but did not implement or test the method. Following Shafer's work, Klinker, et al, [KSK90, Kli93], used the dichromatic reflection model to deduce the skewed-T shaped distribution model, depicted earlier Figure 2, for regions corresponding to single coloured objects under a single illumination. A segmentation algorithm is based around this model. The image is initially divided into equally sized small windows, and segmentation is achieved using a region growing and merging technique – neighbouring regions being merged if they fit the same distribution model sufficiently well. Their algorithm produces a segmentation of the image and separates the body and surface reflection components. Output of the algorithm applied to several images of glossy, brightly coloured plastic objects under a single illuminant is shown, verifying that the algorithm works well in these idealised conditions.

Bajcsy, et al, [BLL96], use the skewed-P shaped colour distribution model, also shown in Figure 2, for their segmentation algorithm used to separate reflection components in an
image. A white reference plate is used to obtain the sensor response to the pure illumination colour and a linear transformation of the colour space is then made so that one of the coordinate axes is aligned with this vector, see section 2.2. Linear clusters of data in the plane defined by the other two co-ordinates are then assumed to belong to single objects. Using this linear model, an image is segmented using a region growing and merging technique similar to that of Klinker, et al. This image is subsequently separated into body and surface reflection components by identifying pixels within segmented regions whose values do not fit a linear model in the three dimensional colour space and labelling such pixels highlight pixels. Again, the algorithm output from images obtained in ideal conditions is shown and looks reasonable.

Healey, [Hea92], uses a unichromatic reflection model to deduce a generic parametric statistical model of the colour data comprising image regions corresponding to single coloured objects in a scene. A binary edge map of the image is produced using a single colour band and, starting with the image as a whole, regions are repeatedly divided into four equal rectangular regions until they contain no edge pixels. A specific instance of the statistical model is then constructed for each resultant rectangular region by choosing values for the parameters of the model such that it best fits the data in the region. If neighbouring regions have sufficiently similar models they are merged into one. The edge detection step implicitly uses a simple region model specifying that regions consist of pixels whose colour values vary smoothly spatially. Thus, as with the early techniques of Ohlander, et al, and Ohta, et al, highlights and shadowed regions will be separated within a region corresponding to a single object. Unichromatic models for differently shaded parts of the same region, however, are similar and so these separated regions get merged in the second stage of the algorithm. The distribution of data in highlight regions generally fits the unichromatic model badly and this fact is used to identify candidate highlight regions. Such candidate highlight regions are merged with surrounding regions if the resultant overall distribution fits a model similar to the skewed-T distribution of Klinker, et al. The output of the algorithm is shown for a few simple images of coloured objects under a single illuminant, which demonstrates efficacy of the algorithm for such scenes. Healey’s statistical model of the data comprising regions of unichromatic reflection is described in section 3.3.

Zhu and Yuille, [ZY96], also use a generic parametric statistical model for the distributions of colour data comprising separate regions based on a unichromatic reflection model to drive their ‘region competition’ segmentation algorithm. Their statistical model is different to the one used by Healey and is also described in section 3.3. The region competition algorithm combines the segmentation techniques of region growing and active contours, both discussed fully in Chapter 4, by using a minimum descriptive length (MDL) approach to minimise the complexity of the segmentation. Further details of the region competition algorithm are given in section 4.5.3. Segmentations generated for a few colour images are shown in [ZY96] and a few more examples are shown in [ZY95]. The environmental conditions
from which the images are obtained are not detailed but appear quite natural, at least one being a daylight scene. Although the theory upon which the algorithm is built is sound, no quantitative analysis of results produced by the algorithm are presented and so it is hard to make comparisons with less sophisticated segmentation algorithms.

Comaniciu and Meer, [CM97], suggest the use of a non-parametric density estimation procedure, the “mean shift” algorithm, which requires no explicit model of the shape of the distributions of data, but infers it from the image. They claim to obtain similar results to Zhu and Yuille’s region competition algorithm but at significantly reduced levels of computational complexity. Once again, however, algorithm output is shown for a few colour images but no quantitative comparison is performed to substantiate this claim.

2.3.2 Object Recognition

Colour has also been used extensively for object recognition. As in colour segmentation algorithms, parametric statistical models of the distributions of colour data corresponding to objects are often used. Generally, in segmentation tasks many unfamiliar objects are encountered and generic models for distribution shapes are required, whereas in object recognition, specific models for different objects of interest are constructed and must be matched to new data. Unlike the models used in segmentation, which tend to be transient instances of generic models for the shape of distributions of colour data and are discarded once a particular task has been completed, models used for object recognition tend to be more permanent.

If the set of objects of interest does not vary widely in terms of the property by which they are being modelled, e.g. colour, then a single permanent model can be used to characterise the entire set. Barni, et al, [BCM97], for example, present a system designed to detect and identify defects in chicken meat. A permanent parametric statistical model of the colour data that comprises good quality meat is used to highlight regions that might correspond to defects. Once defects have been identified, other heuristic rules are used to classify the type of defect. Quantitative results, both in terms of correct overall defective/healthy classifications and confusions between different types of defect, over a test set of 230 images are presented.

Ferri and Vidal, [FV92], use a permanent model in their system, intended to guide a robot fruit picker, which identifies citrus fruit in images also containing leaves and other background artefacts. Their model is a non-parametric statistical model based on a nearest neighbour rule, [DK82]. It is used to classify individual pixels as fruit or background and connected components above a certain size in the resulting binary image are labelled as fruit. The colour model of regions corresponding to fruit is inferred from a training set of two images in which such regions are outlined by hand. The model is used to decide whether each pixel in a test set of four images, in which each pixel has also been classified by hand, corresponds to a
fruit region. Pixel misclassification rates are quoted and used to compare the performance of a number of variants of the nearest neighbour classification rule.

Campbell, et al, [CMT97], use colour as one of the features passed to a neural network at each pixel of an image for classification into one of a number of categories, e.g., road, car, building, sky, vegetation. The neural network is trained on data from images in which each pixel has been classified by hand into one of the predefined set of categories and so encodes permanent models of each region type in terms of the features used. The classification abilities of the model are tested on a large database of images in which regions have also been classified by hand. The use of colour features alone is quantitatively shown to significantly increase classification performance over the use of intensity data alone, and the additional use of texture information is shown to increase overall recognition performance slightly further.

The permanent models discussed so far have been models for a class of object within which there will be a degree of variation. These models therefore are often quite crude and tend only to be useful when applied in very restricted image domains, [Cri92, FV92, BCM97], or in conjunction with other mechanisms or features in larger systems, [CB97, CMT97]. A more common use of permanent object models is when the set of objects to be recognised is finite and those objects are reasonably distinct from each other. In such situations, typically a separate model is stored for each object and when presented with new data, the task is to match that data with an entry in a database of models. Matas, et al, [MMK93], use parametric statistical models for colour data, again presented in section 3.3, to characterise each object in the database. An object model then consists of an instance (i.e., a set of parameter values) of the generic statistical model that fits the data in a training image containing the object. On presentation of the system with an unseen image, an a posteriori probability for each model in the database is calculated for each pixel in the new image and the most likely model for each pixel is found. Groups of spatially proximal pixels with the same most likely model constitute object hypotheses.

Another common model used for colour object recognition is the histogram based model, originally proposed by Swain and Ballard, [SB91]. In [SB91], an object model consists of the histogram of the colour data, represented in the opponent colour space, corresponding to the object in a training image, or possibly a number of training images of the object taken from different viewpoints. Given some new data to match against the set of histograms in the model database, a similar histogram is computed for the new data. Then, for each model, the normalised histogram intersection, given by,
\[
\sum_{j=1}^{n} \frac{\min(I_j, M_j)}{\sum_{j=1}^{n} M_j}
\]

is calculated, where \( j \) enumerates each bin in the 3D colour histogram, \( I_j \) is the count in the \( j \)-th bin of the image histogram and \( M_j \) is the count in the \( j \)-th bin of the model histogram. The object with the model that maximises the normalised histogram intersection is selected as most likely to be observed in the image. Results show high rates of correct matching over a test set in which imaging conditions and geometry are similar to those in the training set.

When the chromatic properties of the illumination change so does the distribution of colour data that occurs in response to an object viewed under that illumination. For a permanent model, this problem must be addressed if objects are to be reliably recognised under more than one type of illumination. Swain and Ballard suggested that all images encountered by their object recognition system might be pre-processed by a colour constancy algorithm, such as those discussed in section 2.1.7, but more elegant solutions to this problem have subsequently been proposed. Funt and Finlayson, [FF95], showed that the ratios of values at neighbouring pixels from each colour band in an image remain roughly constant as the illumination is varied. Exploiting this, rather than modelling objects with the histogram of raw RGB values, they use a derivative of the logarithm of the image and construct histograms of that data for object models. These models are much more robust to changes in illumination chromaticity. Recognition performance was shown to be much better than that obtained with Swain and Ballard's method when there is considerable variation in the illumination, but slightly worse when the illumination is constant due the reduced amount of information contained in the processed image. Nayar and Bolle, [NB96], independently proposed a similar approach and reported good results but do not quote recognition rates.

Matas, et al, [MMK94], construct models from images acquired under each of a number of canonical illuminations. Some consideration of the dichromatic model, neglecting the surface reflection component, leads to the chromatic plane (see section 2.2) as a choice of representation in which to construct object models. A clustering procedure is used to separate the colour histograms into data that arises from different coloured patches on the object. For each object patch, the peak in the corresponding cluster is found in the histogram from each canonical illumination. The convex hull of these peak values serves as a model for each patch ensuring some robustness to varying illumination. Results of application of the algorithm to scenes illuminated by mixtures of indoor lighting and daylight from a window are shown, in which objects from a database are extracted from scenes containing some clutter with reasonable success. Quantitative results are not provided.
Using linear models of surface reflectance and illumination SPDs, as in equations (11) and (12), Healey and Slater, [HS94], showed that the change in colour distribution caused by a change in illumination alone can be modelled by an affine transformation of the RGB colour space. As a result, certain moments of the distributions of colour data arising from the same object under varying illuminations are shown to be invariant and so can be used as a model for that object under any illumination. A small database of object models was constructed from images of objects under white illumination and a test set of images consisted of the same scenes with various colour filters placed over the illumination source. Results show that the invariant models correctly classify every test image. This is a significant improvement over results obtained over the same data set with Swain and Ballard’s method, which correctly classifies about 30% of the test images, and Funt and Finlayson’s method, which recognises about 50% correctly. In later work, [HS97], Healey and Slater also derived invariants for spatially filtered images so increasing the number of features used in object models and increasing overall object recognition performance over an extended test set.

An alternative set of invariants were proposed by Finlayson, et al, [FCF96], and shown to outperform those of Healey and Slater, in terms of recognition rates, on Healey’s extended data set and Swain and Ballard’s data. Rather than an affine model, they restrict the set of colour space transformations caused by an illumination change to diagonal transformations. This set of transformations was shown to be sufficient in earlier work, [FDF94], for representing illumination change. Each component (red, green and blue) image of the colour image is represented as a high dimensional vector containing every pixel value as a separate component. The diagonal transformation model implies that when the illumination changes, these vectors will be scaled but that their orientations do not change. Thus the three angles between these vectors, which are shown to be equivalent to some second order moments of the colour distribution, are illumination invariant. Good performance is obtained by modelling objects with these three values alone, but performance is increased by deriving the same invariants from edge filtered versions of the same image to obtain a further three angles.

Lin and Lee, [LL97], devised a method that does not assume that change in illumination causes an affine transformation of the colour space and so allows for simultaneous variation in both illumination and other factors like object pose which affect the intensity of the colour distribution. They generate binary occupancy grids over the chromatic plane for each database object under a number of canonical illuminants. For each object, a vector is constructed from each 2D grid by concatenating a number of projections of the grid to 1D. A principal components analysis is used to find the major modes of variation of these vectors over the entire database and a manifold that interpolates the vectors corresponding to a single object in the resultant eigenspace serves as a model for that object. The algorithm is shown to work over a small database of objects, but no comparison with recognition rates of other techniques is
provided. Methods like this, and that of Matas, et al, [MMK94], where object models are constructed from multiple images obtained under different lighting conditions, have the advantage that no model for illumination change has to be assumed. Their applicability is however limited to illuminations similar to those in the training images. Methods, such as those of Healey and Slater and Funt and Finlayson, that do assume a model, do not require the large amounts of training data and are potentially applicable to a wider range of illumination conditions.

No spatial information is encoded in any of the colour object models discussed above but such information can be encoded into a model by using a graph representation. Another limitation of histogram models is that in order for matching to occur, the region of the new image to be matched against the models in the database must take up most of that image, or be extracted from that image. Otherwise the histogram model of the new image may become confounded by background clutter making matching difficult. Region adjacency graphs (RAGs), introduced by Syeda-Mahmood, [Sye97], represent an object as a set of distinct regions defined by their colour and size, and a set of adjacencies between regions. An unseen image is represented similarly in its entirety and object identification and recognition is achieved by sub-graph matching. Extraction of relevant regions before matching is therefore not necessary, but, for complex imagery, the sub-graph matching process may be computationally expensive. Matas, [Mat96], introduced a similar concept, the colour adjacency graph (CAG), in which an object is represented only by the set of distinct colours that neighbour each other rather than including every distinct region. The colour adjacency graph encodes a less strict topological model of the object than the region adjacency graph and so is less sensitive to changes in viewing conditions, particularly non-rigid changes of shape. In order to extract all colour adjacencies from a database object, the union of colour adjacency graphs obtained from images of the object from different angles can be used. The graphs also contain reflectance ratio values, c.f., [FF95, NB96], with each adjacency to decrease sensitivity to changing illumination. Matas shows that CAG models achieve a high recognition rate over a small database of objects but no comparison is provided with the performance of other types of model. Some favourable comparison of the compactness and computational efficiency of CAG models compared to RAG models is given.

2.3.3 Tracking

Modelling distributions for discrimination between different regions of an image, as required for segmentation, is a simpler task than modelling for identification as required for object recognition. In object recognition, the models need to be permanent rather than transient and so generally they need to be appropriate in a much wider variety of environments. The tracking problem is a mixture of discrimination and identification – initially a model must be constructed
to discriminate the object of interest from the background and in subsequent frames of the image sequence, the object must be identified using that model. In a typical tracking task however, consecutive frames of the sequence are obtained under very similar conditions and so the degree of generality required of the object models used is not as great as for object recognition.

Thorpe, et al, [THK88], use a k-means algorithm to extract four clusters out of two sets of sample data, one from the object (road) and one from the background (non-road), in the colour road tracking component of their autonomous vehicle. Each pixel in a new image is classified as road or non-road by finding the cluster from the two models to which it is most likely to belong. The image region corresponding to the road is assumed to be triangular and the most likely parameters of this shape are deduced from the classified image by use of a Hough transform. Their statistical model is continuously reconstructed from the regions classified as road and non-road in each image. A buffer zone around the boundary between the two regions is excluded to minimise the effects of small errors made on the previous frame. It is assumed that seed regions containing data of each type are given at start-up. Good performance, in terms of failure of the overall system, which contains other components based on texture and stereo information, is reported. However, in the sample images shown, the road is quite distinct from the background and it is not clear how well the technique would fare for less well-defined objects of interest. Furthermore, the continuous update of the statistical model is a lengthy process and the reported work rate of the system is slow – the vehicle moves at 10cm/s.

Crisman, [Cri92], further developed the colour road tracking system of Thorpe, et al. She simplifies the system by using a permanent parametric statistical model, again comprised of four Gaussian clusters extracted from sample data using the k-means algorithm. The set of roads on which the system is trained and tested is very restricted, consisting of images of paths in a university campus all made of the same material and having similar surroundings. This allows successful application of this permanent road colour model. Crisman's colour model is applied to images containing these paths and a likelihood is assigned to each pixel that it belongs to the path. The resultant likelihood image is compared to hierarchically stored, binary template road images, some of which contain branches and forks, and the best match selected. No direct quantitative assessment of the colour model is provided but good results of the overall algorithm in terms of robustness of the navigation system that uses it are reported.

Crowley and Berard, [CB97], incorporate a permanent skin colour model as one component in their robust face tracking system. Over a training set of pixels known to correspond to skin regions a histogram is computed and normalised by the size of the training set. New data is given a probability of corresponding to a skin region by examining the corresponding entry in the normalised histogram. This crude skin colour model is used for initial detection of regions in the image likely to correspond to a face. No analysis of the
performance of the colour model itself is performed, but overall performance of the tracking system is reported to be high and, in particular, robust to failure of individual components.

Brock-Gunn and Ellis, [BE92], use models similar to the histogram models of Swain and Ballard for tracking and identification of targets through sequences of images from a stationary camera. Data corresponding to the target is obtained by subtraction of a background image in which no targets are in view and then matched against a database of existing models. Models are stored hierarchically, i.e., at different histogram resolutions, to speed up the matching search. Matching speeds are compared favourably with a simple linear search, but matching success rates are not quoted. Rehg, et al, [RLW97], take a very similar approach in their 'smart kiosk'. As well as models being required for tracking individuals allowing the eyes of the virtual salesman's face to follow people as they move around, the models are stored so that people can be recognised and not approached more than once.

Active contour models, originally proposed by Kass, et al, [KWT87], and discussed in Chapter 4, are a useful tool for extracting contours and boundaries from images and have proved effective in tracking applications. Although more commonly applied to grey level intensity images, they have been applied successfully to multiband, in particular colour, images by Sapiro, [Sap96], Ngoi and Jia, [NJ96], Onyango and Marchant, [OM96], and Ivins and Porrill, [IP95, Ivi96]. In the work of Ivins and Porrill, for example, objects of interest are tracked through sequences of images using a technique based on active contour models, which exploits a model of the colour data comprising the region of interest. An instance of a generic parametric statistical model is constructed from a seed region contained in the region of interest, which is assumed to be given at the outset. That model is used to characterise the region throughout the sequence but is subsequently discarded, a new model being created for each individual tracking task. The statistical models used by Ivins and Porrill are similar to those used by Matas, et al, [MMK93], and are also discussed in section 3.3. The other works mentioned above do not use an explicit model of the region of interest but seek edges in the colour images to which they are applied, [Sap96, NJ96], or use colour features as part of a model for specific landmark points in a region, [OM96]. Further details of all these works are given in Chapter 4.

2.3.4 Content-Based Image Retrieval

Related to object recognition is the use of colour for retrieval of relevant images from databases. Typically in object recognition a database of object models is stored and new data consists of an image possibly containing one of those objects. In image retrieval a query is somehow specified and a database of images must be searched for matches consisting of images that somehow relate to that query. Das, et al, [DRD97], describe the FOCUS system for searching large databases of colour images. Queries are in the form of image regions, containing for example a product logo or distinctive animal markings, and images with similar content are sought by
searching each image of the database for similar regions. A candidate image list is generated as quickly as possible by matching histogram peaks between the queried image region and each image of the database. A spatial proximity graph of each candidate image is constructed, in which histogram peaks that are spatially proximal in the image are connected, and compared to a similar representation of the query. This eliminates images in the candidate list that contain the same colours as the query region but at very different locations in the image. This image representation is similar to Matas’s colour adjacency graph but is somewhat simplified to reduce storage size and computation over a large database. Several sample queries are shown which produce a good set of matching images, and quantitative results over a test database are quoted in terms of the precision and recall rates on 25 test queries. The precision rate examines the percentage of matched images that are deemed accurate by a human overseer, and the recall rate is the percentage of database images deemed relevant to the query that are recalled by the system. A similar system is described by Huang, et al, [HKM97]. Their system is based around several colour image features derived from the spatial correlation of distinct colour values. Matches between these correlation values are sought in the query and database images. Examples are shown and performance is assessed in a similar way to Das’s system but over a separate image database.

The obvious application of this type of system is for stand alone image or video libraries, but another potential domain is the world-wide web. The use of such systems for retrieval of images from the web could have important implications for the vision community particularly with regard to the acquisition of test sets of images for system development. It is questionable however if low-level features, like colour, provide the best basis for such systems particularly at the user interface level. Other methods, used for example in the Web-Seer system of Swain, et al, [SFA97], combine some image analysis, designed to make broad distinctions such as photograph/non-photograph, with a text query matched against the html page on which the image was found.

2.3.5 Other Uses

Colour data can also be used for edge detection (see for example [Pra78]) for which it is supposed that more reliable results should be produced than for intensity data alone by virtue of the extra-dimensionality. However, it is commonly reported, see [Luo93] and references therein, that little advantage is gained by the use of colour over intensity images. The colour based active contour models of Sapiro, [Sap95], and Ngoi and Jia, [NJ96], also use colour for edge detection. Some improvement over grey level data is claimed for these techniques, however, these claims tend to be based on inspection and are not supported by any hard, quantitative empirical evidence.
Novak and Shafer, [NS94], showed that certain scene properties, such as phase angle and surface roughness, can be estimated from the shape of the histogram of colour data of a single coloured object under a single collimated illumination. The dichromatic reflection model is assumed, with neutral interface reflection; Lambertian reflection is assumed for body reflected light and the Torrance and Sparrow, [TS67], model for surface reflected light. The mathematical expressions that define the reflection models for each component prove difficult to invert and so, although some measurements taken from the colour histogram can be predicted from scene parameters, it proves difficult to derive analytic expressions that predict scene parameters from histogram measurements. The models are used instead to find a number of corresponding values for scene parameters and histogram measurements and a look-up table, which quadratically interpolates between these points, is used to relate histogram measurements to scene parameters. Novak and Shafer’s algorithm separates an observed colour histogram into two body and surface reflection clusters using a skewed-P shape model. Measurements of the length and width of the surface reflection cluster and the point of intersection of the body and surface reflection clusters are made. The look-up table is then used to find corresponding values of the roughness of the surface, the intensity of the illumination and the phase angle, between the illumination and viewing directions. The surface defined by the look-up table is shown not to self-intersect so that unique values for the scene parameters exist for each combination of histogram measurements. Results compare reasonably well to ground truth data for estimation of the illumination intensity and phase angle. For surface roughness, where the ground truth is hard to obtain, a number of different surfaces are shown to be ranked in order of roughness in the same way as they were ranked by a human observer.

2.4 Discussion

The review in the preceding section identifies a number of different ways in which objects can be modelled in terms of the colour data that corresponds to them:

- Parametric statistical models: the parameters of a generic probability distribution function are chosen so that its shape best fits a sample set of data. Measures of likelihood for new data can be derived directly from the probability distribution function. A relatively small amount of sample data is required to construct an effective model provided the generic shape is appropriate. Although these models can be arbitrarily complex, many distributions can be modelled reasonably well by fairly simple density functions, which are simple and computationally cheap both to fit to sample data and to extract measures of likelihood from for new data. The disadvantage of these models is that in any non-trivial application, the model simplifies reality and will never fit a true distribution perfectly. Models of this type have frequently and successfully been used for tracking, as in [THK88, Cri92, IP95].
Chapter 2 Colour

- Non-parametric statistical models:— no shape is enforced on these models, but the structure of the distribution is inferred from the sample. Non-parametric models will almost always outperform parametric models provided there is sufficient sample data available, [Rip96], but when the size of the sample is limited, the enforced shape of a parametric model can often prove advantageous. These models often have minimal construction time, as no function has to be fitted to the sample data, but generally more work is required to obtain measures of likelihood. This latter point may often be ameliorated, however, by judicious use of a look-up table (see for example, Appendix B, section B.2). These models could be used in place of any parametric model used in a tracking application, but probably would be most effective when the class of objects of interest varies sufficiently little to allow the use of a permanent model, c.f., Crisman's road model. In such an application, the model is not constructed on-line and so it is easy to ensure that sufficient training data is provided. Although only tested on single images, the non-parametric model of Ferri and Vidal, [FV92], described in the previous section, is designed for just such an application.

- Histogram models:— histogram models are a simple form of non-parametric model and similar arguments apply. For a histogram model, the likelihood assigned to new data depends solely on the bin count of exactly similar data over the sample and so a large set of sample data must be provided to ensure efficacy of the model. Although histogram models have been used in tracking applications, [BE92, RLW97], the actual tracking task in both these situations is made somewhat trivial by using a fixed camera and stationary background. The models are used more for the purposes of recognising individual objects of interest than to assist the tracking process. For less constrained tracking tasks, where sample data must be provided at start-up, the use of histogram models may not be advisable.

- Neural Networks:— A neural network can provide a measure of likelihood that new data is of a particular class in much the same way as a statistical model and so could be used in the same way for a tracking application. However, it is only practical to use a neural network as a permanent model, for example, a neural network similar to the classifier of Campbell, et al, [CMT97], could be used in the place of Crisman's parametric statistical model. The large training times required make online construction of transient neural network models impractical.

- Graph-based models:— These models also require a large amount of data to be reliably constructed. Furthermore, they are designed to model objects that consist of a number of distinct regions. Although for such objects these models could doubtless be used for tracking, particularly if permanent models for the objects could be constructed a priori, the objects of interest here tend to be of essentially one colour and do not exhibit strong spatial structure.
Chapter 2 Colour

Although many different types of model exist, the review in the last section highlights a distinct lack of empirical comparison between both the different types of model and different models of the same type. The only real comparisons that have been performed have been amongst the object recognition community, and most of these comparisons are concerned with what data should be used rather than how it should be modelled. The simplicity and robustness of parametric statistical models of colour data make them an appealing choice for tracking applications and such models are used in the remainder of this thesis. No comparison with other types of model is attempted and attention is focussed on the identification of the most appropriate parametric statistical model for natural distributions of colour data. As discussed above, several models have been suggested before, some of which are based on simple physical models such as the unichromatic reflection model, [Hea92, MMK93, Ivi96, ZY96], and some of which simply use very flexible and general models, [THK88, Cri92, IP95]. In the next chapter, these models are reviewed and some alternatives are presented. One of the main goals of this thesis is to provide a framework for empirical comparison of these models and thus to generate a comparison of a set of candidate statistical models for natural distributions of colour data.
Chapter 3 Statistical Modelling

Statistical models are often an essential component of systems, like many vision systems, which encounter large amounts of noisy data. Given a new piece of measurement data, a statistical model supplies a measure of likelihood that the entity from which the measurement was taken is the same as the one that is being modelled. In order to construct a statistical model, some sample measurements from the entity for which the model is required must be obtained. However, an effective statistical model not only captures the shape of this sample distribution of data, but also reflects the parent distribution from which that sample was drawn. To ensure good generalisation to unseen data, a statistical model must be as close as possible to this parent distribution. In this work, the entities being modelled are regions in sequences of images corresponding to objects of interest in a scene and the measurement data is colour camera data. Thus, an appropriate statistical model for characterising the distributions of colour data contained in such regions is sought. Figure 3 shows two images each containing a region of interest for which a statistical model might be required.

![Figure 3 Two images containing regions of interest that need to be characterised by a SCM. The seed region is outlined in each image.](image)

In the image on the left, the region of interest is that corresponding to the green shorts (the white pillow case is also a potential region of interest but for now only the green shorts are considered) and on the right, the country lane. Superimposed on each image is a polygon defining a seed region, which is used to provide a sample set of data from which a model can be constructed. Each image is the first from a sequence containing the same object of interest. In the sequence containing the shorts, the position of the object varies with respect to the background but the camera and light source positions are roughly fixed (the camera is hand held). The physical conditions in which this sequence was obtained adhere closely to the assumptions of the dichromatic reflection model, i.e., there is a single illuminant and the object of interest is of one colour. Moreover, the object of interest is matte, exhibiting little surface reflection, and so a unichromatic model of reflection may be sufficient. The object is non-rigid ensuring a wide range of orientations of surface patches with respect to the light source and
camera positions. In the other sequence, the illumination is daylight and the object of interest is made of gravel, which may have shiny surfaces causing surface reflection. The orientation of the surface patches is somewhat restricted, as the object is approximately planar. The assumptions underlying a unichromatic model of reflection are less well satisfied for these images, but the model may still provide a reasonable approximation. The camera position moves steadily down the track as the sequence continues.

In Figure 4 occurrence histograms of the data contained in the seed regions, corresponding to the sample distributions, from each image in Figure 3 are shown as well as the occurrence histograms of colour data corresponding to the region of interest as a whole throughout each image sequence. The latter histograms approximate the parent distributions of colour data that correspond to the regions of interest in the environmental conditions under which these images were obtained. However, no idea of the density of the distribution is conveyed by these images, which are generated by filling each position of the histogram at which the measurement count is non-zero.
Chapter 3 Statistical Modelling

This chapter presents several models for such naturally occurring distributions of colour data and concludes with a discussion of how the most appropriate might be chosen. For each model, it is assumed that the object of interest is essentially of one colour (i.e., a single intrinsic surface reflectance, \( s(\lambda) \)) with small variations superimposed on it and that the illumination, \( i(\lambda) \), also has approximately constant chromaticity. The corresponding distributions of data thus form a single cluster in the measurement space, which here is the RGB colour cube. All the models are parametric statistical models, which consist of a probability distribution function whose exact shape is specified by some tuneable parameters. Construction of these models from a sample distribution of data is generally performed by maximum likelihood estimation, which seeks to find the values for the parameters that maximise the value of the probability density function over the sample distribution.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GNCC</td>
<td>3.3.1</td>
<td>2D Gaussian model in the normalised colour space.</td>
</tr>
<tr>
<td>OPC*</td>
<td>3.3.2</td>
<td>Oriented Planar Chromaticity model. A Gaussian distribution in the plane tangent to the unit sphere at the point corresponding to the mean direction in the sample distribution.</td>
</tr>
<tr>
<td>ZYC</td>
<td>3.3.3</td>
<td>Zhu and Yuille’s Chromaticity model. A Gaussian model of the residual vectors of each data point from a line passing through the origin in the direction of the mean of the sample distribution.</td>
</tr>
<tr>
<td>ANC*</td>
<td>3.3.4</td>
<td>Angular Normal Chromaticity model. A Gaussian model of the angles of longitude and colatitude required to represent the direction of rays from sample data point to the origin.</td>
</tr>
<tr>
<td>BC*</td>
<td>3.3.5</td>
<td>Bingham Chromaticity model. A Bingham model of the directions of the lines joining each sample to the origin.</td>
</tr>
<tr>
<td>R1BC*</td>
<td>3.3.6</td>
<td>First Robust Bingham Chromaticity. The contribution of each sample to the matrix that determines the shape of the Bingham model is weighted by its intensity.</td>
</tr>
<tr>
<td>R2BC*</td>
<td>3.3.6</td>
<td>Second Robust Bingham Chromaticity. The contribution of each sample to the matrix that determines the shape of the Bingham model is weighted by the square root of its intensity.</td>
</tr>
<tr>
<td>HC</td>
<td>3.3.7</td>
<td>Healey’s chromaticity model. The trivariate Gaussian model of the sample data projected onto the unit sphere by normalising the mean vector and covariance matrix.</td>
</tr>
</tbody>
</table>

Table 1 Summary of directional chromaticity models. * indicates that a chromaticity model has not been used previously and so is an original contribution of this thesis.
Chapter 3 Statistical Modelling

Apart from the trivariate Gaussian model presented in section 3.2, all the models exploit the independence of the chromaticity and intensity components of colour data. As they are independent, these two components can be modelled separately and the models later combined to obtain a full colour model. Several models for the chromaticity component are presented in section 3.3. Table 1 provides a summary of these chromaticity models and indicates which of them have been used previously and which are original contributions of this thesis. Models for the intensity component are discussed in section 3.4. Table 2 summarises these models and again indicates which models are new. The abbreviations given in the tables are described in the text of the corresponding sections.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UI</td>
<td>3.4.1</td>
<td>Uniform Intensity model.</td>
</tr>
<tr>
<td>GI</td>
<td>3.4.2</td>
<td>Gaussian Intensity model.</td>
</tr>
<tr>
<td>PSI*</td>
<td>3.4.3</td>
<td>Planar Surface Intensity model.</td>
</tr>
</tbody>
</table>

Table 2 Summary of intensity models. * indicates that an intensity model has not been used previously and so is an original contribution of this thesis.

3.1 Statistical Colour Models

A statistical model characterises some entity by modelling the distribution of values of some measurement taken from that entity. In order to put these models to practical use, they must provide, when presented with some new measurement data of the same type, a value indicating the level of certainty that the entity from which the measurement was taken is the entity characterised by the model. For statistical colour models (SCMs), given a new piece of colour data – an RGB vector, the model returns a measure of likelihood that a pixel with that colour vector belongs to the image region being modelled. In this section, a general definition of the SCMs used here, in terms of the measures of certainty they provide, is given. Two such measures are used throughout this work, the “goodness” and the “likelihood”. Both these measures are defined in terms of a distance function, $d : \mathbb{R}^n \rightarrow \mathbb{R}_+$, where $n$ is the dimension of the measurement data – 3 for RGB colour data. This distance function provides a measure of the distance or difference between some new piece of data and the sample used to construct the statistical model. The form of the distance function is derived from the probability distribution function of the particular model being used. Once the distance function is defined, then the goodness, $G : \mathbb{R}^3 \rightarrow (-\infty,1]$, is given by,

$$G(R,G,B)=1-\frac{1}{k}d(R,G,B).$$

(17)

$k (>0)$ is a parameter that is introduced into the SCM to control the tolerance of the model. As $k$ is increased, so is the goodness assigned to any new piece of data (unless the distance is zero in
which case the goodness is always one). The goodness is defined in such a way that data that, according to the model, are similar to the sample data have a positive goodness, but for data further from the sample, the goodness decreases eventually becoming negative and tending to minus infinity. As $k$ is increased, more data gets assigned positive goodness. This goodness function follows the definition used by Ivins and Porrill, [IP94, Ivi96].

The likelihood, $L : \mathbb{R}^3 \rightarrow (0, 1]$, can be defined in terms of the goodness as

$$ L(R, G, B) = e^ {G(R, G, B) - 1} = e^ {- \frac{1}{k} d(R, G, B)} . $$

The definition of the likelihood still contains the tolerance parameter $k$ and as $k$ is increased, the likelihood of any new piece of data increases similarly. For data close to the sample distribution, the likelihood is close to one; and as data becomes further away, the likelihood tends to zero. This definition is similar to the likelihood function used by Crisman, [Cri92], but incorporates the tolerance parameter used by Ivins and Porrill.

### 3.2 Gaussian models of colour data

A very common statistical model is provided by the multivariate Gaussian distribution. The Gaussian distribution is a very flexible model, it is simple and easy to compute and, in practice, it models many naturally occurring distributions of data at least reasonably well. The probability density function for the multivariate Gaussian is

$$ N(x; \bar{x}, S) = \frac{1}{\sqrt{(2\pi)^n |S|} } \exp \left[ - \frac{1}{2} (x - \bar{x})^T S^{-1} (x - \bar{x}) \right] $$

where $n$ is the dimension of the data being modelled, $\bar{x}$ is the mean of the distribution and $S$ is the covariance matrix. The mean vector and covariance matrix completely define the model and, thus, are the parameters for which estimates are required from samples. Suppose there are $m$ pieces of data, $x_i, i = 1, \ldots, m$, in the sample distribution. Maximum likelihood estimates of the mean vector, $\bar{x}$, and covariance matrix, $\tilde{S}$, [DH73], are given by,

$$ \bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i $$

and

$$ \tilde{S} = \frac{1}{m} \sum_{i=1}^{m} (x_i - \bar{x})(x_i - \bar{x})^T . $$

The Mahalanobis distance, $M$, of some new piece of data, $x$, from the estimated mean of the model is defined by

$$ [M(x, \bar{x})]^2 = (x - \bar{x})^T \tilde{S}^{-1} (x - \bar{x}) $$
Chapter 3 Statistical Modelling

and gives a measure of distance of any new piece of data from the sample distribution, [DH73].

For colour data, $x=(R, G, B)$, so $n=3$ and the $x_j$ are the values of the pixels in the seed region. Thus the multivariate Gaussian SCM is obtained by replacing the general distance function, $d$, in (17), with the Mahalanobis distance from the mean, given in (22), thus

$$d(x) = \sqrt{(x - \bar{x})^T S^{-1} (x - \bar{x})}.$$  \hspace{1cm} (23)

Figure 5 shows a fixed likelihood boundary of Gaussian models of the two regions of interest in Figure 3, constructed from the sample distributions shown in Figure 4, the ellipsoidal shape of the multivariate Gaussian distribution is evident in both cases. The boundary shown, as with all the subsequent pictures of SCMs, is the zero goodness boundary with the tolerance parameter set at the value found empirically, see Chapter 7, to minimise the total number of misclassifications.

Figure 5 Gaussian models constructed from the sample distributions shown in Figure 4.

Trivariate Gaussian models for RGB colour data have been used in the past. Crisman, [Cri92], and Thorpe, et al, [THK88], model clusters of colour data with Gaussian distributions; in both cases, however, multiple clusters are allowed. Ivins and Porrill originally suggested the use of the Gaussian model for whatever data comprised the image to which their active region model was applied, [IP94, IP95]. Although often this data was multiband MRI/PET data, [IP94a], or texture data rather than colour data, a trivariate model of colour data was used in the work presented in [IP95]. In that work, it is suggested that using in a Gaussian model in the HSV space, but omitting the value (c.f., intensity) component, might be preferable. The parametric statistical model of pixel colours corresponding to non-defective chicken meat used by Barni, et al, [BCM97], is also a trivariate Gaussian.
3.3 Modelling the Chromaticity Component

Although the multivariate Gaussian distribution can often be a reasonable default model, by examining the physical processes that give rise to a distribution of data, it is often possible to devise more accurate alternatives. In Chapter 2, the physical processes underlying the acquisition of colour data were investigated and in this section, that analysis is used to derive some alternative models for colour data. A unichromatic model for the reflection process is adopted, as expressed in equation (2), section 2.1.1. Following Matas et al, [MMK94], and Zhu and Yuille, [ZY96], it is assumed that data with a non-negligible surface reflection component is sparse and so its effects simply become one of the components of noise in the data. If the effects of noise are ignored for the moment, the sensor response to a particular surface patch in a scene is constrained to lie on a line in the RGB colour cube that passes through the origin. Thus, the chromaticity under a particular illumination can be characterised by the direction of that line or equivalently by a unit vector within that space. Over an extended surface, the surface reflectance at each point is drawn from some distribution in the space of SPDs. Suppose the reflectance, \( s(\lambda) \) at a point on an essentially single coloured surface is drawn from a Gaussian distribution, i.e.,

\[
\begin{align*}
    s(\lambda) &\sim N(\bar{s}(\lambda), C(\lambda_1, \lambda_2))
\end{align*}
\]

where \( \bar{s}(\lambda) \) is the mean surface reflectance and \( C \) is a covariance function in which \( \lambda_1 \) and \( \lambda_2 \) both index the wavelength of light. The distribution of image irradiance chromaticity then also has Gaussian characteristics. As shown in section 2.1.2, when a unichromatic reflection model is adopted, the direction of the ray from the origin to a data point represents the chromaticity of that data. Thus, adopting this Gaussian model of surface reflectance, assuming linearity of the camera the chromaticity distribution of the observed data, or equivalently, the distribution of directions of the rays joining data points to the origin, can also be expected to have a Gaussian profile.

There are a number of ways in which such a distribution can be modelled using directional statistics, [Mar72, Mar88, FLE87], and the rest of this section presents several alternatives. Throughout this section, no knowledge of the intensity component is assumed and so a uniform model of the distribution of intensities is adopted. Alternative models for the intensity component to the data are discussed later in this chapter, in section 3.4.

Figure 6 shows fixed likelihood boundaries of SCMs based on a directional chromaticity model and a uniform intensity model for each region of interest in the images of Figure 3. The likelihood along lines passing through the origin is constant and so the distribution has a conical shape with an elliptical cross section and apex at the origin. Figure 7 depicts the sample and parent chromaticity distributions, where the chromaticity of each data
point is represented by the intersection of the ray, from the origin to that point, with the unit plane in the RGB cube. The unit plane in the RGB cube is the triangular region, in the RGB cube, with vertices at (1,0,0), (0,1,0) and (0,0,1) – bottom left, top, and bottom right, respectively, in the figures, as shown by the coloured outlines. In Figure 7, the intensity at each position indicates the frequency with which the corresponding chromaticity occurs.

Figure 6 Directional chromaticity, uniform intensity SCMs constructed from the sample distributions shown in Figure 4.

Figure 7 Sample (top) and parent (bottom) chromaticity distributions, from the images of Figure 3, shown in the unit plane.
3.3.1 Gaussian Models in the Unit Plane

As shown in Figure 7, one way to represent the direction of a line passing through the origin of a 3D space is by a point on a fixed plane, for example the unit plane, within that space. In [IP95], Ivins and Porrill suggested the use of a 2D Gaussian model of the hue and saturation components, from the HSV colour space, as a model for distributions of colour data. The non-linear nature of the transformation between the RGB and HSV spaces, however, makes this choice of representation unwise for modelling distributions of data, as it is not clear how this transformation affects the shape of the distribution. This is later acknowledged by Ivins, [Ivi96], who uses 2D Gaussian distributions in the normalised colour space instead. As mentioned in section 2.2, the normalised colour space is equivalent to the unit plane in the RGB cube, as normalised colour co-ordinates are obtained by taking the $L_1$ norm of an RGB vector. The unit plane is a reasonable choice of fixed plane in which to represent directions, as the ray from the origin to every non-zero point in the RGB colour cube must intersect this plane. Matas, et al, [MMK93], used similar 2D Gaussian models in the CIE chromaticity plane, which is equivalent to the plane $B=1$ in the RGB cube, see section 2.2. This choice of fixed plane is less wise as the chromaticity of data for which $B=0$ is not represented.

Following Ivins, to generate a fixed plane Gaussian model, the two normalised colour co-ordinates, $r$ and $g$, from equation (14), in section 2.2 (page 33), are used and a 2D Gaussian model is constructed, as described in section 3.2. The distance function, defining the Gaussian normalised colour (GNC) chromaticity model is the Mahalanobis distance from the mean, given in (23). Figure 8 depicts the GNC chromaticity models in the unit plane constructed from the sample distributions shown in Figure 7. In these pictures, the intensity at each position indicates the probability, according to the model, that a pixel with corresponding chromaticity is part of the region of interest. This applies to all similar images shown later in this chapter.

Figure 8 Gaussian chromaticity models in the unit plane.
3.3.2 Oriented Planar Models

For simply characterising the chromatic properties of single items of colour data, the use of the unit plane is sufficient, as the $L_1$ norm is a perfectly valid representation of a direction, but there are problems when trying to characterise distributions of chromatic properties. When the mean of a distribution of directions is close to one of the co-ordinate axes, the distribution of points of intersection with the unit plane is spread wider than a similar distribution whose mean is aligned with the normal to the plane. This effect is illustrated in 2D in Figure 9, where two directional distributions with similar variance are shown. The range over which intersections with the unit line occur is much greater for the distribution that is not aligned with the normal to that line. A Gaussian model of distribution 2, in Figure 9, on the unit line will have greater variance than that of distribution 1, and is therefore a less tight model of the true distribution.

![Figure 9](image)

**Figure 9** Demonstrates how tighter models of directional distributions can be constructed by choosing a plane perpendicular to the mean of the distribution.

The directions of lines passing through the origin can be represented as points in any plane that intersects each positive axis of the RGB space and does not pass through the origin. A consistently tight model can be ensured, as illustrated in Figure 9, by choosing a plane whose normal is aligned with the mean direction of the distribution. The distribution of directions can then be modelled in that plane by modelling, as before, the distribution of points at which the rays from the data points to the origin intersect the plane.

Suppose the estimated mean direction is given by the unit vector 
\[
\hat{\mathbf{x}} = \frac{1}{\sqrt{R^2 + G^2 + B^2}} (R, G, B),
\]
then two perpendicular unit vectors lying in the plane perpendicular to $\hat{\mathbf{x}}$ are:
\[
\hat{\mathbf{y}}_1 = \frac{1}{\sqrt{R^2 + G^2}} (-G, R, 0), \text{ and }
\]
\[
\hat{\mathbf{y}}_2 = \frac{1}{\sqrt{R^2 + G^2} \sqrt{R^2 + G^2 + B^2}} (-RB, -GB, R^2 + B^2).
\]

Given a piece of data $\mathbf{x}$, the point of intersection of the ray from that point to the origin with the plane perpendicular to the mean direction (arbitrarily, but without loss of generality, chosen to
Chapter 3 Statistical Modelling

be at a distance of 1 from the origin) is \(\left(1/x \cdot \tilde{x}\right)\tilde{x}\). The direction, or chromaticity, is thus represented by the point

\[
\tilde{x}_p = \left(\frac{1}{\tilde{x} \cdot \tilde{x}}\right) \tilde{x}_p (x, y) \cdot \tilde{x}.
\] (26)

The mean direction, \(\tilde{x}\), can be found by taking a unit vector in the direction of \(\tilde{x}\) in (20).

A 2D Gaussian model of these points of intersection is constructed and a distance function that defines the Gaussian oriented planar (GOP) chromaticity model is again given by the Mahalanobis distance from the mean of this distribution, see (23). Figure 10 depicts GOP chromaticity models, in the unit plane, constructed from the sample distributions shown in Figure 7. For both models, the range of chromaticity for which the probability is non-negligible is smaller than for the GNC chromaticity models shown in Figure 8, which indicates that a tighter fit is achieved.

![Figure 10 Gaussian chromaticity models in planes oriented with the mean of the sample distributions. Shown in the unit plane.](image)

3.3.3 Zhu and Yuille's SCM

Zhu and Yuille, [ZY96, ZY95], suggest a model for characterising distributions of colour data that is similar to the oriented planar models of the last section. From the sample data, a suitable mean direction, \(\tilde{x}\), and intensity, \(\tilde{I}\), are chosen by assuming that each measurement, \(x\), consists of a chromatic component, given by a unit vector, \(\tilde{x}\), scaled by the illumination intensity, \(I\), and a residual noise component, \(e\). Thus,

\[
x = I \tilde{x} + e.
\] (27)

\(\tilde{x}\) and \(\tilde{I}\) are then determined by minimising the residuals in the least squared sense, i.e.,
Chapter 3 Statistical Modelling

\[
(\tilde{x}, \tilde{I}) = \arg\min_{\tilde{x}, \tilde{I}} \sum_{i=1}^{m} |x_i - L^i \tilde{x}|^2 = \arg\min_{\tilde{x}, \tilde{I}} \sum_{i=1}^{m} |e|^2.
\]

Gradient descent or singular value decomposition are suggested by Zhu and Yuille to perform the minimisation and obtain an estimate of the mean direction, \(\tilde{x}\). In this thesis, however, \(\tilde{x}\), of (20), is used as a simple approximation, as in the previous section. Once \(\tilde{x}\), has been calculated, the residuals of each data point from that mean direction lie in a plane perpendicular to \(\tilde{x}\). This plane is defined in exactly the same way as the oriented planes in the GOP model and the residuals are modelled by a 2D Gaussian distribution in that plane. The Zhu and Yuille (ZY) chromaticity model is then defined, as usual, by the Mahalanobis distance from the mean as in (23).

Figure 11 ZYCU1 models constructed from the sample distributions shown in Figure 4.

The difference between this model and the oriented planar model of the previous section is that, for this model, colour vectors are not normalised before the residuals are calculated in either the model construction phase or when calculating likelihoods for new data. This has two effects on the model: firstly, higher intensity data contributes much more to the shape of the model than low intensity data in the model construction phase. Secondly, all lines parallel to \(\tilde{x}\) are now lines of constant likelihood rather than all lines passing through the origin of the RGB cube as they are for the other directional models. Assuming a uniform model for the intensity component of the data, as adopted by Zhu and Yuille, this gives the model a cylindrical shape, still with elliptical cross-section, as opposed to the conical shapes of the other directional models that have been discussed. These models, for the two regions of interest in Figure 3, are shown in Figure 11. Zhu and Yuille argue that down-weighting low intensity data in this way is beneficial because the chromatic component of low intensity data is very sensitive to noise because of the poor resolution in the dark corner of the RGB cube.
Chapter 3 Statistical Modelling

Problems exist with all these planar representations of direction. When a planar representation is used, the distance between a variable pair of directions that are fixed with respect to each other is not consistent. A more natural and consistent representation for the direction of a ray from the origin is by the \( L_2 \) unit vector in that direction, or, equivalently, by a point on the unit sphere. The chromaticity distribution observed in response to a surface whose reflectance at a point is drawn from a Gaussian distribution may then be expected to have Gaussian profile on the unit sphere. When such a distribution is projected onto the unit plane, it no longer has Gaussian profile. The change of shape becomes more drastic as the mean direction of the distribution moves away from the direction perpendicular to the unit plane and the distribution on the plane becomes asymmetric and skewed and so is not modelled well by a Gaussian distribution. This effect is illustrated well by Healey, [Hea92], but can be seen by considering Figure 9 in which the unit circle is shown.

When the mean and the normal to the plane are well aligned, the corresponding distribution on the plane is at least symmetric and so a Gaussian model is more appropriate. The two vectors in (25), together with the point \( \vec{x} \), in fact describe the tangent plane to the unit sphere at the estimated mean direction. When the distribution is tight about the mean, the range on the unit sphere over which the probability is non-negligible may be small enough to be well approximated by the tangent plane and so oriented planar models can be an effective approximation. However, it is possible to obtain more accurate models for these distributions on the unit sphere directly and such models are described in the remainder of this section.

3.3.4 The Angular Normal Distribution

Another common way to represent a point on the unit sphere is by the angles of longitude, \( \theta \), and colatitude, \( \phi \). This is exploited in the familiar spherical polar representation, \((r, \theta, \phi)\), of position in a 3D space, see for example [Kre93], where \( r \) denotes Euclidean distance from the origin. Given a distribution of directions assumed to have Gaussian profile on the unit sphere, a naïve way to construct a model might be to construct a 2D Gaussian model of these two angular co-ordinates. Such models have been used in the literature, for example, as a model of the distribution of surface facet orientations used by Torrance and Sparrow, [TS67], and followed by Novak and Shafer, [NS94]. There are, however, two significant problems with models of this type.

Firstly, the finite nature of the spherical surface means that, particularly for models with large variance, there may be a significant wrapping effect, where the tails of the distribution still have non-negligible weight after having gone right around the sphere. There is a well established distribution on the circle, the wrapped normal distribution, [FLE87, Mar72], which is constructed in exactly the way suggested in the last paragraph, but with just one angle to
Chapter 3 Statistical Modelling

represent direction. The wrapped normal distribution accounts for the wrapping effect by summing over all possible angles that indicate the same direction, i.e., angles obtained by adding any integer multiple of $2\pi$, the probability density function is thus given by,

$$\text{WN}(\theta; \bar{\theta}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{r=-\infty}^{\infty} \exp \left[ -\frac{(\theta - \bar{\theta} + 2r\pi)^2}{2\sigma^2} \right]$$

where $\bar{\theta}$ is the mean of the distribution and $\sigma$ is the covariance, both calculated by maximum likelihood estimation in the usual way. A similar expression, involving a double summation could doubtless be derived for a similar distribution on the sphere. However, when the variance is small, as for the distributions of colour data under investigation here, contributions to the summation from angles outside the $[0, 2\pi]$ range are negligible and can be considered zero. The probability density function then reduces to that of the familiar 2D Gaussian distribution, which, if $\Theta$ to denotes the vector $(\theta, \phi)^T$, is given, c.f., (19), by

$$\text{AN}(\Theta; \bar{\Theta}, S) = \frac{1}{2\pi|S|} \exp \left[ -\frac{1}{2} (\Theta - \bar{\Theta})^T S^{-1} (\Theta - \bar{\Theta}) \right].$$

The distance function, defining the angular normal (AN) chromaticity model, is given, as usual, by the Mahalanobis distance from the estimated mean of the model, as in (23).

![Angular normal chromaticity models depicted in the unit plane.](image)

The second and more serious problem with this model concerns a topological feature of the sphere that causes this model to have distinctly non-Gaussian profile particularly near the poles of the sphere where $\phi$ tends to zero. In this region of the sphere, large changes in $\theta$ have only a slight effect on the corresponding position on the sphere and thus distributions tend to become uniform with respect to $\theta$. When the distribution being modelled is very tight about the mean and the mean is not close to the pole, this distribution may provide a reasonable model, but, in general, it should be used only with extreme caution. Angular normal models for the two images in Figure 3 are depicted in Figure 12. Both models are tight about their means, neither of
which is in the vicinity of the pole of the sphere, and so the non-Gaussian profile of the model is not evident. The models appear similar to the corresponding planar models shown in Figure 10, but when they are viewed consecutively to highlight difference, the angular Gaussian models can be seen to be slightly tighter than the Gaussian oriented planar models.

### 3.3.5 The Bingham Distribution

A well established spherical analogue of the planar Gaussian distribution, for which topological features of the sphere are accounted for, is the Bingham distribution, [Bin74, Mar72, Ken87]. The Bingham distribution has probability density function

\[
B(\hat{x}; \mu, \kappa) = \frac{1}{4\pi d(\kappa)} \exp \left\{ \text{tr} \left( \kappa \mu^T \hat{x} \hat{x}^T \mu \right) \right\}
\]

where \( \mu = (\mu_1, \mu_2, \mu_3) \) is an orthonormal matrix describing the centre \( (\mu_3) \) and major \( (\mu_2) \) and minor \( (\mu_1) \) axes of the elliptical contours of the distribution on the unit sphere, \( \kappa = \text{diag}(\kappa_1, \kappa_2, \kappa_3) \) is a matrix of constants describing the ratios of the lengths of the major and minor axes (\( \kappa_3 \) is in fact always scaled to zero), and \( d(\kappa) \) is a constant depending only on the values of the non-zero elements of \( \kappa \) - see [Bin74, Mar72]. The maximum likelihood estimate of \( \mu \) is the matrix of eigenvectors of the sample second moment matrix

\[
T = \frac{1}{m} \sum_{i=1}^{m} \hat{x}_i \hat{x}_i^T
\]

and the estimates of the \( \kappa_i \) are non-trivial functions of the eigenvalues of \( T \), involving confluent hypergeometric functions, see [Bin74, Mar72, Ken87]. For the Bingham distribution, a measure of distance from the distribution is given by

\[
d(x) = \sqrt{-\text{tr} \left( \kappa \mu^T \hat{x} \hat{x}^T \mu \right)},
\]

and this defines the Bingham (B) chromaticity model.

Bingham chromaticity models for the sample distributions shown in Figure 7 are depicted in the unit plane in Figure 13. These models are visibly tighter than the planar models shown in Figure 8 and Figure 10, and the angular normal models shown in Figure 12. As the model distributions are tight about their means and those mean directions are quite close to the normal of the chromaticity plane, the non-Gaussian profile (i.e., non-elliptical contours) caused by projection from the unit sphere to the chromaticity plane is not evident from Figure 13.

The Bingham distribution actually models axial data and so exhibits antipodal symmetry on the sphere, so \( B(\hat{x}) = B(-\hat{x}) \). This property of the distribution is redundant in this application, however; as RGB values cannot be negative, all the data resides on the positive octant of the sphere.
3.3.6 Coping with Noise

As in Zhu and Yuille's SCM, it is possible to down-weight noisy data close to the origin of the RGB colour cube in the other directional models. For example, the shape of the Bingham model may be influenced by noise components in data close to the dark corner of the cube. Typically, the effect of noise is to add or subtract one or two quantisation levels to one or more of the three bands of a colour image. In the vicinity of the origin this can have a significant effect on the direction of the ray from the origin to a measurement. As the intensity is increased however, and the value moves further from the origin, the effects of noise on the direction of the line are very much reduced. Thus it seems reasonable that higher intensity data should be relied upon more when determining the shape of the chromaticity model. By decreasing the influence of low intensity data on the shape of the chromaticity models accuracy may be enhanced. This can be achieved for all the models discussed above by weighting the contribution of sample data by a function of the intensity. For the Bingham model, for example, when calculating the sample second moment matrix, $T$, see (32), from which all the shape parameters are calculated, the contribution of each sample can be weighted by its intensity to obtain a more robust sample second moment matrix,

$$ T_R = \frac{\sum_{i=1}^{m} \hat{X}_i \hat{X}_i^T |X_i|}{\sum_{i=1}^{m} |X_i|} . $$

The calculation of the parameters of the Bingham distribution then proceeds as before and the distance, defining the first robust Bingham (R1B) chromaticity model, is defined exactly as in (33). R1B chromaticity models for the samples in Figure 7 are shown in Figure 14. Little difference between the standard Bingham models of Figure 13 can be seen from these pictures,
however, when viewed consecutively the wider robust model for the green shorts can be seen to be very slightly tighter, but the track model appears identical to the standard Bingham model.

Figure 14 Robust Bingham models depicted in the unit plane.

In (34), the factor by which data is down-weighted is a linear function of the intensity of the data, which by not normalising the colour vectors when constructing their model, is effectively the down-weighting function that Zhu and Yuille use. This choice of down-weighting function can be justified by a simple calculation if the noise is modelled as an additive isotropic Gaussian distribution. A fixed percentage of the measurements, \( x \), obtained from a constant image irradiance will then lie within a sphere centred at some mean value, \( \overline{x} \).

Figure 15 Disparity of directions of lines from the origin caused by uncertainty in the absolute position of a data point.

Suppose that one such sphere is taken as the range over which measurement values that arise in response to constant image irradiance can vary. With reference to Figure 15, the position that maximises the difference in the direction of the line joining the measurement to the origin must occur when \( x \) lies on the plane perpendicular to the line joining \( \overline{x} \) to the origin and at the extremes of the sphere. If \( c \) is the radius of the sphere, then the maximum difference in direction, or uncertainty of the direction, can be described by the angle \( \theta \). From the diagram,

\[
\sin \theta = \frac{c}{X}.
\]
Chapter 3 Statistical Modelling

For small $\theta$, $\sin \theta \approx \theta$ and so the range over which the direction, $\theta$, can vary (i.e., the uncertainty) therefore depends on the inverse of the intensity, $|x|$. It seems reasonable to down-weight contributions by the inverse of the uncertainty, which is linear in the intensity.

Experiments have also been performed using a down-weighting function dependent on the square root of the intensity and these models look very similar to those depicted in Figure 14. The expression used for the sample second moment matrix is similar to that given in (34), but the intensity, $|x|$, is replaced by its square root. The chromaticity model constructed in this way is referred to, henceforward, as the second robust Bingham (R2B) chromaticity model.

### 3.3.7 Healey’s SCM

Healey, [Hea92], based a SCM used for image segmentation on a similar physical model of the underlying image formation processes. Healey removed the intensity component from the SCM by $L_2$ normalising the parameters of a trivariate Gaussian model constructed from the seed region, effectively obtaining a trivariate Gaussian model of the distribution of $L_2$ unit vectors in the directions of the sample data. The mean of the model is found, in exactly the same way as the mean direction is found for the oriented planar models of section 3.3.2, by normalising the mean of a trivariate Gaussian model. The covariance matrix is found by normalising the trivariate covariance matrix by the squared magnitude of the mean. Measures of likelihood for new data are then found by using the Mahalanobis distance, exactly as in (23), between the unit vector towards the new datum and the normalised mean of the model.

![Figure 16 Chromaticity models constructed using Healey’s approach, depicted in the unit plane.](image)

By initially constructing the model in the RGB colour cube rather than directly on the unit sphere, an inherent robustness to noise is introduced into the model and the down-weighting suggested in the previous section for the other directional models is not necessary. Chromaticity models constructed in this way from the samples shown in Figure 7 are depicted in Figure 16. The model for the green shorts is significantly tighter than the Bingham based models indicating that the effects of noisy data near the origin of the RGB cube are very much
Chapter 3 Statistical Modelling

reduced. This model appears to be a closer approximation to the sample distribution from the shorts shown in Figure 7, but it is not clear which model is closest to the parent distribution. For the track, however, the model appears slightly wider than the Bingham models, although it does appear tighter than the angular normal and planar models.

3.4 Modelling the Intensity Component

If the direction of the line joining a sensor data point to the origin of the response space characterises the chromatic properties of the reflected light that produced that response, the distance along that line from the origin characterises the intensity. The unichromatic model of reflection implies that these two components of the reflected light are independent and this allows them to be modelled separately. In the last section the directional chromatic component was modelled. In this section the intensity component is discussed.

As the two components are independent, then if \( p_c(x) \) and \( p_i(x) \) are the probability density functions of the models of the chromatic and intensity components respectively, then the probability density function for the combined model is given by

\[
p(x) = p_c(x)p_i(x).
\]

The distance function is typically obtained by taking the logarithm of the probability density function, ignoring constant additive terms and taking the square root, thus the overall distance functions can be separated into a term for each of the chromatic and intensity models:

\[
d(x) = \sqrt{d_c^2(x) + d_i^2(x)}.
\]

Several different representations for the chromaticity component are used for the different models in the previous section and different definitions of the intensity, \( |x| \), that are consistent with the individual models will be adopted. For the spherical models – the Bingham, angular normal and Healey models, the intensity is given by

\[
|x| = \sqrt{R^2 + G^2 + B^2}.
\]

For the unit plane model, however, which is based on the L_1 norm rather than the L_2,

\[
|x| = R + G + B
\]

For the other models, the oriented planar and Zhu and Yuille models, the intensity definition in (39) must be generalised to

\[
|x| = x \cdot \tilde{x}.
\]
### 3.4.1 Uniform Intensity Model

Suppose the object of interest is some highly non-rigid body such as the shorts in Figure 3. For such a surface, there is little knowledge available about the distribution of orientations of the facets that make up that surface and, in particular, that distribution may vary considerably from the distribution that gave rise to the sample data. In such circumstances, no knowledge about the distribution of intensities can be assumed and the wisest approach may be to adopt a uniform model for the distribution, i.e., all intensity values are presumed equally likely. The distance function for this uniform model is constant over the RGB cube and so can be omitted from the overall distance function. Thus, all the distance functions corresponding to the different chromaticity models presented in the previous section implicitly make this assumption, as no intensity component is included in those expressions.

### 3.4.2 Gaussian Intensity Model

Now suppose the object of interest is more rigid and that its shape does not change much throughout a sequence of images. Under these circumstances it may be possible to constrain the intensity model somewhat. A simple restriction, with little physical justification other than the fact that there can be many factors affecting the distribution of intensities, as noted in section 2.1.5, is to use a Gaussian model of the intensity. Such a model has been shown to be an improvement over the uniform model in certain circumstances, [AB97]. The Gaussian distribution has historically been widely used to model distributions of intensity data corresponding to image regions arising from separate objects in a scene, [ZM90, Ron94, ZY96], though it is generally recognised that a better model could probably be found.

During the model construction phase, the intensity of each sample is used to construct a 1D Gaussian model by maximum likelihood estimation. The probability density function of the resultant complete colour model, as shown in (36), is found by multiplying together the probability density functions of the chromaticity and intensity models. Using a Bingham chromaticity model for example, the probability density is given by

$$BCGI(x; \eta, \sigma, \kappa, \mu) = \frac{1}{\sigma \sqrt{2\pi} \sqrt{4\pi d(\kappa)}} \exp \left[ \frac{\text{tr}(\kappa \mu^T \hat{x} \hat{x}^T \mu)}{2\kappa^2} - \frac{(|x| - \eta)^2}{2\sigma^2} \right]$$

where $\eta$ and $\sigma$ are the mean and standard deviation of the Gaussian intensity model respectively. Thus the following distance function, in the form of (37), which defines the Bingham chromaticity, Gaussian intensity (BCGI) SCM, is obtained:

$$d(x) = \sqrt{\frac{(|x| - \eta)^2}{2\sigma^2} - \text{tr}(\kappa \mu^T \hat{x} \hat{x}^T \mu)}.$$
Figure 17 shows fixed likelihood boundaries of a BCGI model and a Zhu and Yuille chromaticity model with a Gaussian intensity (ZYCGI) model of the two regions of interest in Figure 3.

The ZYCGI models are ellipsoidal and appear similar to the trivariate Gaussian models shown in Figure 5. The main difference between these two types of model is that, unlike the trivariate Gaussian, the major axis of the ZYCGI model is constrained to lie along a line that passes through the origin. The BCGI models have a different shape, which widens further away from the origin. Combining a Gaussian intensity model with any of the other directional chromaticity models results in a model with similar shape to the BCGI model.

3.4.3 Exploiting Shape Knowledge

In section 2.1.3, the dependence of the shape of the body reflection cluster on the object surface shape was discussed. In particular, when the unichromatic reflection model is adopted so that the body reflection cluster is the only cluster, the shape of the surface of the object of interest
may be used to derive information concerning the distribution of intensities more accurately. One of the classes of objects of interest here is lanes and tracks in daylight scenes. These objects are approximately planar and in this section, this knowledge is used to derive an alternative intensity model. As mentioned in section 3.3.4, both Torrance and Sparrow, [TS67], and Novak and Shafer, [NS94], model surfaces as a collection of planar facets whose orientations are modelled locally by an isotropic angular normal distribution. If the whole surface is roughly planar, such a model can be extended to the surface as a whole. As the intensity of the reflected light is highly dependent on the angle between the surface normal and the incident light direction, such a model may provide insight into the distribution of intensities that can be expected from these surfaces.

Figure 18. Rough surface illuminated and viewed from fixed directions

Consider the idealised viewing scenario depicted in Figure 18. Suppose that the direction of the illumination incident on the surface is constant and that the direction of the normal to each surface facet is drawn from some symmetric distribution of directions, e.g., a Bingham distribution. The distribution of angles of incidence, $\alpha$, is then also symmetric and so may be modelled by a Gaussian distribution.

From the unichromatic reflection model, specified in equation (2), section 2.1.1, if $L(\lambda)$ is the scene radiance, then

$$|L(\lambda)| = m_b(g)|s(\lambda)i(\lambda)|.$$  \hspace{1cm} (43)

$m_b$ is some function of the scene geometry $g$. If $m_b(g)$ is replaced by some constant multiple of the cosine of $\alpha$, the simple Lambertian model of equation (8), section 2.1.3, is obtained. If this model is assumed, the intensity is a non-linear function of the angle of incidence:

$$\alpha = \cos^{-1}\left(\frac{|L(\lambda)|}{c|s(\lambda)i(\lambda)|}\right).$$  \hspace{1cm} (44)

where $c$ is a constant, and so the Gaussian distribution is not preserved. As the variance increases, the expected distribution of intensities becomes skewed towards higher intensities. $c|s(\lambda)i(\lambda)|$ in equation (44) is the intensity of the scene radiance at a point for which the angle of
incidence is zero and so is the maximum possible image irradiance. As \( \alpha \) is drawn from a Gaussian distribution, equation (44), predicts that the distribution of inverse cosines of the ratio of scene radiance to the maximum scene radiance also follows a Gaussian distribution.

For a linear sensor, the shape of this distribution in the response space can be expected to be similar. Thus, for a distribution of measurements, \( \mathbf{x} \), taken from some planar surface, the inverse cosine of the intensity, \( \cos^{-1}(\frac{|x|}{\text{max} I}) \), scaled by the maximum possible intensity value, \( \text{max} I \) say, can be modelled by a Gaussian distribution giving the following probability distribution function:

\[
p(\cos^{-1}(\frac{|x|}{\text{max} I}) = \mu) = \frac{1}{Z} \exp\left\{-\frac{1}{2} \left( \frac{\cos^{-1}(\frac{|x|}{\text{max} I}) - \mu}{\sigma} \right)^2 \right\}.
\]  

(45)

where \( Z \) is the constant of normalisation.

A further parameter is now introduced into the model, which is the value of \( \text{max} I \). \( \text{max} I \) should have the value of the intensity of the data corresponding to the maximum intensity light that is reflected by the surface, i.e., when \( \alpha = 0 \). Although a lower bound on \( \text{max} I \) is available, as it must be greater than or equal to all values of the intensity found in the sample data, it can potentially take any higher value and is not limited to the range of sensitivity of the camera. Novak and Shafer, [NS94], discuss the use of the length of the body reflection line as a colour histogram measurement from which to estimate scene parameters. Measuring the length of the body reflection line requires knowledge of \( \text{max} I \), and Novak and Shafer simply assume that the largest measurement in the sample corresponds to that value. In their application, this assumption is reasonable since their objects of interest are spherical and so the full range of angles of incidences occurs, provided the illumination direction is within 90° of the viewing direction. Here the objects under consideration are planar and so this luxury is not afforded. Attempts to derive an analytic expression for the maximum likelihood estimate of \( \text{max} I \) from the sample data have not been successful and so a more explicit approach has been adopted. The intensity model is constructed repeatedly, using different settings for the \( \text{max} I \) parameter, until a maximum likelihood over the sample is found. The algorithm used to find the maximum likelihood setting of \( \text{max} I \) is given in Appendix BSection B.1.

This model is combined with chromaticity models in exactly the same way as other intensity models and, again using the Bingham chromaticity model as an example, the probability density function of the overall model is given by
Chapter 3 Statistical Modelling

\[
BCPI(x; \eta, \sigma, \kappa, \mu) = \frac{1}{Z_{4\pi d}(x)} \exp \left[ \text{tr} \left( \kappa \mu^T \hat{x} \hat{x}^T \mu \right) - \cos^{-1} \left( \frac{|x|}{\max |x|} - \eta \right) \right].
\] (46)

The corresponding distance function is

\[
d_{BCPI}(x) = \sqrt{\left( \cos^{-1} \left( \frac{|x|}{\max |x|} - \eta \right) \right)^2 - \text{tr} \left( \kappa \mu^T \hat{x} \hat{x}^T \mu \right)},
\] (47)

which defines the Bingham chromaticity planar surface intensity (BCPSI) model.

Figure 19 Sample (top) and parent (bottom) distributions of intensities from the objects in Figure 3 with uniform, Gaussian and planar intensity models. As usual, the pictures on the left correspond to the green shorts and those on the right to the track.
Figure 19 shows how the different intensity models discussed above fit the real distributions of intensities found in the regions in the images of Figure 3. The planar surface intensity models do not appear to differ greatly from the Gaussian model. For the green shorts, it is hard to discern which of the models provides the best fit. For the track, the non-uniform models appear to be preferable. There appears to be a slight bi-modality in the sample intensity distribution for the track, which neither of the non-uniform models can capture, and it is difficult to decide which of the two models is closer. The track parent distribution, however, does not appear bimodal and exhibits some skew towards higher intensity values. The planar model appears to be a slightly better fit to the track parent distribution than the Gaussian. Constant likelihood boundaries of full SCMs incorporating this model look very similar to those for SCMs incorporating Gaussian intensity models, shown in Figure 17.

3.5 Other Models

In this section some other models, which may be useful for modelling colour data, but which have not been investigated due to the constraints of time are mentioned.

3.5.1 Multivariate t models

The multivariate t-distribution is a distribution with elliptical contours of constant probability, like a Gaussian, but with longer tails (decaying according to an inverse power law) than the Gaussian distribution. The effect of the longer tails is to downweight outliers in training samples. As described by Lange, et al, [LLT89], modelling with t-distributions rather than normal distributions is very much in the spirit of robust statistics. Careful studies have shown that real distributions often do not quite follow normal distributions but have slightly longer tails, [Rip96, LLT89], and so are better modelled by a t-distribution.

In a full working system, the seed region would need to be extracted by some automatic procedure and, consequently, some of the sample data may not actually be part of the region of interest. In addition, when such a system is applied to real world situations, the SCM would need to be regularly updated in order to cope with changing environmental conditions. The data used to reconstruct the model must be taken from current estimates of location of the region of interest, which may, as it has been automatically extracted, contain errors and thus introduce outliers into the sample distribution. If more robust statistics are used, outliers are less likely to irreparably corrupt the statistical model and utilising a t-distribution rather than a normal distribution would increase the tolerance to errors in the seed region extraction process. The t-distribution could replace the Gaussian in any of its applications in previous sections of this chapter - trivariate models, 2D planar models, or 1D intensity models.

The family of multivariate t distributions contains a degrees of freedom parameter, $v$, [Rip96, LLT89]. As $v \to \infty$, the t distribution approaches the normal, and the smaller the value of...
v the longer the tails of the distribution. In this way, v may be regarded as a robustness parameter, which can be tuned by hand or, if there is sufficient data available may be estimated along with the other parameters of the distribution by maximum likelihood estimation, as described in [LLT89].

3.5.2 Kent's distribution

As already noted in section 3.3.5, the Bingham distribution exhibits antipodal symmetry. Although this property is completely redundant here, as the data being modelled only occurs in the positive octant of the unit sphere, it might be preferable to use a distribution that is designed to model distributions with just one mode on the sphere. In the practical book on directional statistics by Fisher, et al, [FLE87], Kent's distribution, [Ken82], is used to model single mode distributions on the sphere. Fisher in fact avoids the use of the Bingham distribution for modelling distributions of this type, because of difficulties in obtaining measures of goodness of fit from Bingham models. However, both Bingham's and Kent's distributions are a 5-parameter special cases of the more general, 8-parameter, Fisher-Bingham distribution, also described in [Ken82], and consequently have very similar form and profile.

3.5.3 The Angular Central Gaussian distribution

The angular central Gaussian distribution, analysed in detail by Kent and Tyler, [Tyl87, KT88], and also referred to as the offset normal, [Mar72], is another symmetric distribution on the sphere and so may be used to model symmetric distributions of directions. The angular central Gaussian distribution is obtained by projecting a zero mean Gaussian distribution onto the unit sphere, [KT88], and thus is antipodally symmetric like the Bingham distribution. It is deemed worthy of mention by virtue of the close relationship it has with the circular wrapped Cauchy distribution, which has similar profile in 1D — for details see [KT88]. In much the same way as the multivariate t distribution, of section 3.5.1, the Cauchy distribution can be used as an alternative to the normal distribution that provides some robustness to outliers by virtue of heavier tails. In a similar way, the angular Gaussian distribution provides a more robust model for distributions of directional data. This could be beneficial in two ways, firstly by down-weighting noisy data close to the origin of the RGB colour cube as discussed in section 3.3.3. Secondly, using a more robust model leaves the statistical model less sensitive to inaccuracies in the seed region as discussed in section 3.5.1. The Cauchy distribution does not have the tuneable degrees of freedom parameter of the multivariate t and so it, as well as the related angular central Gaussian distribution, is not as rich as the full family of t distributions. However, no spherical analogue of the t-distribution is known to the author.
3.5.4 Poisson Intensity Model

Pal and Pal, [PP91], suggested that the Poisson distribution should be used as a model for the distribution of intensities found in homogeneous image regions. This model is based on the observation of Dainty and Shaw, [DS74], that over regions of a sensor array where the exposure is constant, the number of photons being captured per second in each bin follows a Poisson distribution. In this thesis it is assumed that the variation in the intensity caused by material or geometric variation of the surface of the object of interest is sufficiently large that effects like these can be considered negligible. This model is has not been considered further.

3.6 Limitations

The directional SCMs described above are derived from a very simple physical model, which by no means includes all the factors that can contribute to give naturally arising distributions of colour data their shape. This section serves as a reminder of other factors, not taken into account by these models other than to add to the overall noise component, that affect the shape of real distributions.

3.6.1 Surface Reflection

A physical model of the surface reflection component of light reflected from a surface is not explicitly incorporated into the directional statistical colour models. It is assumed that data containing a non-negligible surface reflection component is sparse and so does not contribute significantly to the shape of the overall distribution. Often however, there may be a considerable surface reflection component in the overall distribution especially, for example, in outdoor scenes if the surface of a road or lane is wet.

3.6.2 Colour Constancy

No attempt is made to make the statistical models colour constant, i.e., robust to changes in illumination chromaticity. It may be possible to construct statistical models that are less sensitive to changes in illumination by modelling less sensitive colour features, for example, local colour ratios, [FF95, NB96]. Using such a scheme would almost certainly raise misclassification rates when the illumination is constant but help when the illumination changes in an unpredictable way. Although in the unconstrained outdoor environment in which an ALV might work, illumination conditions can change unpredictably, so may the colour of the surface being tracked. Colour constancy clearly cannot help with the latter situation and so, rather than attempting to use a model that is robust to changes in environment, simpler models are used, which, if necessary, could be frequently updated to keep track of changing statistics of the region of interest.
3.6.3 Lighting components

As discussed in section 2.1.5, for daylight scenes the illumination conditions are more complex than those that are assumed in the unichromatic model used to derive the SCMs above. In that section, it was suggested that, since the illumination provided by the sky is reasonably uniform over a planar surface such as a road, the addition of a constant term to the reflection model may be appropriate for daylight scenes. Incorporation of such a term has the effect of translating, away from the origin, the point at which the lines representing chromaticity equivalence classes converge. For a non-ideal surface for which, rather than being constant, the reflectance at a point is drawn from a Gaussian distribution, the component of the reflected light corresponding to this constant illumination is also drawn from a Gaussian distribution. Incorporation of this constant lighting component complicates the structure of the statistical models and has not been attempted in this thesis. It is worth noting that when there is no directional component to daylight, i.e., the sky is overcast and there is no direct illumination from the sun, see section 2.1.5, the only term in the reflection model is the constant term. In this type of situation, the trivariate Gaussian model of section 3.2 may prove to be a very good approximation to real distributions of colour data.

3.6.4 Intensity Dependent Chromaticity

The directional models rely heavily on the assumption that as the viewing geometry is varied, although the intensity of the reflected light may change arbitrarily, the chromatic properties are constant. For a wide range of materials this assumption is valid, see section 2.1.4, but also for some it is not. For many materials, for example cloths like velvet, the chromaticity of the reflected light is highly dependent on the direction from which they are viewed. For such materials the models discussed here are not appropriate and alternatives, possibly based on the BRDF, would be required, some of these materials are discussed in the work of Dana, et al. [DNV97].

3.6.5 Dependence of Intensity on Viewing Direction

In the derivation of the planar surface intensity models in section 3.4.3, the commonly adopted Lambertian model for body reflected light is assumed. Several experimental studies have shown this model to be an over simplification when dealing with natural materials and more comprehensive models have been proposed, for example by Wolff. [Wol96]. However, in general, the Lambertian model is considered to be a reasonable approximation when the viewing and incidence angles are not close to grazing.
3.6.6 Multi-Modality

All of the models discussed in this chapter are strictly single mode, and so are only applicable when the object of interest has a single mode distribution of surface reflectance SPDs and is viewed under an illumination whose spatial chromaticity distribution also has a single mode. When the object of interest is comprised of different materials with different chromatic properties, i.e., it is multicoloured, or when there are stark changes in illumination chromaticity in a scene, the distribution of colour data observed in response to the object can become multimodal. However, in these more complex environments, the same basic laws of reflection used to develop the single mode models still apply. Given a sample set of data from an object of interest with a number of distinct, differently coloured components, a clustering procedure is required to separate the sample into clusters corresponding to distinct components of the object. Each cluster can be viewed as a sample from a single coloured object of interest and modelled with any of the models described above. The resultant set of single mode statistical models must then be somehow combined to form a multimode statistical model.

The original intention for this project was to develop multimode models in this way once the most suitable models for single mode distributions had been identified. Although some preliminary work towards this has been performed, some problems were encountered and the models have not been fully tested. Details of this preliminary work are however included as an appendix (Appendix A) to this thesis.

3.7 Summary

In this chapter a number of existing directional chromaticity models have been discussed and some new alternatives proposed. Some models for the intensity component have also been proposed. The intensity component is assumed to be independent of the chromaticity, which allows intensity and chromaticity models to be combined easily. The set of chromaticity models that have been implemented and tested during this project is summarised at the beginning of this chapter in Table 1 and the intensity models are summarised in Table 2.

In the remainder of this thesis, the abbreviations given in Table 1 and Table 2 are concatenated to provide the name of a complete model. For example, BCGI denotes the SCM with a Bingham chromaticity model and a Gaussian intensity model. Use of these abbreviations will be avoided wherever possible but is essential for some of the graphs and tables of results given in Chapter 7.

3.8 Conclusions

A number of different models for distributions of digital colour data have been proposed. Pictures of these models have been provided wherever possible to assist visualisation of the
differences between the models. Comparison of these pictures with real distributions can provide some idea of the suitability of different models, but, in order to choose the best model reliably, a quantitative framework for comparison must be developed. There are three broad classes of measurement that can be made on these models to quantify their integrity:

- Goodness of fit to the sample distribution.
- Goodness of fit to the parent distribution.
- Classification performance.

Methods for obtaining measures of goodness of fit of a model to the sample from which it is constructed are available in the literature and comparison of such measurements can be used to select the best model. A naïve approach is to select the model that maximises the likelihood of the sample. However, problems with this scheme arise when there are hierarchies of models. A model hierarchy exists when one model is a special case of other higher dimensional models. Higher dimensional models will always fit a finite sample distribution better than lower dimensional, special case, models even if the lower dimensional model is a perfect fit for the parent distribution. However, higher dimensional models are more vulnerable to the influence of artefacts of the sample distribution, which are not present in the parent distribution, and may therefore generalise less well to unseen data. An example of such a nested hierarchy of models is the set of intensity models proposed in section 3.4. As the variance of the Gaussian model tends to infinity, it approximates the uniform model. Furthermore, although no formal proof of equivalence has been attempted, by carefully selecting the parameters of the planar intensity model, very close approximations to Gaussian distributions can be obtained.

There are measures, such as Akaike’s AIC (“an information criterion”), [Aka74], and the minimum descriptive length (MDL) measures of Rissanen, [Ris78], that take account of the dimension of a statistical model and so can be used to select the most appropriate from a nested hierarchy. Problems remain however, as these measurements still rely on the likelihood assigned to the sample distribution. Often, application specific knowledge can incorporated into the model construction phase to find parameter settings that do not correspond to the maximum likelihood settings, but that may provide a better fit to the parent distribution. In this chapter, for example, the robust Bingham models of section 3.3.6 incorporate the application specific knowledge that data closer to the origin is less reliable than more distant data to find an alternative set of parameter values. The likelihood of the sample distribution assigned by the standard Bingham model will always be higher than that assigned by the robust Bingham model and so it will always be selected when using measurements based on likelihood of the sample. It is reasonable to expect, however, that by basing estimates on more reliable data, the robust Bingham model might provide a better fit to the parent distribution.
Chapter 3 Statistical Modelling

Despite the problems discussed above, measures of goodness of fit to the sample distribution, such as the AIC, do have distinct advantages. In particular, they can be computed directly from a given sample distribution and so used to choose from a set of candidate models on-line. If a measure of this type could be shown to reliably select the most appropriate model, for example, by comparison with choices based on a more trustworthy measurement, it could be a very useful tool. In this thesis, however, attention is directed towards development of the more reliable measure and results have not been compared with those obtained from these simple measures.

The second type of measurement assesses the goodness of fit of a model to data, drawn from the parent distribution, that was not included in the sample from which the model was constructed. Some such measures, for example cross-validation, [Rip96], hold back some sample data from the model construction phase and test models on this unseen sample and so can be used to select the most appropriate model on-line if sample data is abundant. The model that maximises the likelihood of the unseen sample is selected as best. When sufficient data is available, in particular when a distribution approximating the parent distribution is available, such methods provide a good idea of which model generalises best to unseen data. The integrity of these measurements might be increased by further examining the likelihood assigned by models to negative cases, i.e., pixels from outside the region of interest, in order to gain some idea of the classification potential of the model.

In this thesis, however, the third approach is adopted where the classification performance of each model is assessed directly and used to compare the models. In a typical application of the statistical models discussed in this chapter, the model is used to classify pixels as in or out of the region of interest. The model itself does not perform this classification directly; it simply provides measures of likelihood that are used by a classification algorithm. By measuring the performance of a classification algorithm, over a test set of data, using different underlying models, a comparison of the models themselves can be obtained. Such an approach is appealing, as it is unlikely that one model can be exclusively identified as the best for modelling all distributions of colour data (or any other type of data). The best choice of model may even vary depending on the choice of classification algorithm. This approach allows the most appropriate model for a particular application, using a particular classification algorithm, to be identified. One disadvantage of the approach, as will become apparent in later chapters, is that classification algorithms, particularly more complex ones, may introduce their own artefacts into the performance measures that are independent of the type of underlying statistical model. Such artefacts can bias the comparison of the models. Furthermore, this type of approach can be a lengthy process involving many experiments requiring large amounts of computation time.
As discussed in the introduction, the application of interest in this thesis is tracking of regions of interest through sequences of images. A region tracking algorithm can be viewed as a classification algorithm that classifies each pixel of each image of a sequence as in or out of the region of interest. The following two chapters examine the region tracking algorithms themselves and attempt to find an algorithm that is both effective for tracking the regions of interest in this work (roads/lanes and items of clothing), and that introduces few of its own artefacts into performance measures taken from it.
Chapter 4 Region Segmentation and Tracking

The application for which the statistical colour models developed in the last chapter are intended is the tracking of regions of interest through sequences of images. In order to perform this task, an algorithm is required that can exploit a statistical model in order to classify each pixel of each image in a sequence as in or out of the region of interest. There are many existing techniques for extracting regions from images or sequences of images and this chapter reviews these techniques in an effort to develop algorithms that might be used in conjunction with the statistical models of Chapter 3.

4.1 Definitions

Hoover, et al, [HJJ96], remark that definitions of image segmentation are not consistent in the image processing and computer vision literature. They extend the definition in Gonzalez and Woods, [GW92], where an image segmentation is defined as follows:

“Let R represent the entire image region. We may view segmentation as a process that partitions R into n sub-regions, R₁, R₂…Rₙ, such that

1. \( \bigcup_{i=1}^{n} R_i = R \),
2. \( R_i \) is a connected region, \( i=1,2,…n \),
3. \( R_i \cap R_j = \emptyset \) for all \( i \) and \( j \), \( i \neq j \),
4. \( P(R_i) = \text{TRUE} \) for \( i=1,2,…,n \), and
5. \( P(R_i \cup R_j) = \text{FALSE} \) for \( i \neq j \).

Where \( P(R_i) \) is a logical predicate over the points in set \( R_i \) and \( \emptyset \) is the null set.”

Levine, [Lev85], and Ballard and Brown, [BB82], suggest that item 5 above should apply only to adjacent regions as non-bordering regions may well have the same properties. This suggestion is adopted by Hoover, et al. In the pattern recognition literature, see for example, [Rip96, DK82], a standard inclusion for classification procedures, such as segmentation, is the possibility of a doubt category to which cases about which there is too little information to make a definite decision are assigned. Hoover, et al, who are dealing with segmentations of range images, use the similar notion of a non-surface pixel and note that in general a region corresponding to this category may not be connected and thus violate item 2 of the definition above. Item 2 is omitted in the definition of segmentation given by Levine, [Lev85]. Haralick and Shapiro, [HS92], do not attempt a formal definition but resort to informal rules.

The full segmentation problem is only of peripheral interest here and no attempt is made to resolve these issues. The restricted problem of region segmentation is of more direct interest.
Chapter 4 Region Segmentation and Tracking

The objective of a region segmentation algorithm is to extract a single region from an image and, thus, a region segmentation is a segmentation in which there are just two categories, object and background, and possibly a doubt category. Although it may often be satisfied, the connectedness restriction, item 2 in the definition above, is abandoned here to allow for the possibility that the positively classified (object) region is not connected or contains holes, as is often the case with simple algorithms based on thresholding techniques, discussed in the next section. A region segmentation is defined to be an image in which each pixel is classified as object, background or doubtful. A region segmentation algorithm takes any image as input and outputs a region segmentation.

The definition of a region tracking procedure is a simple extension of region segmentation. A region tracking algorithm takes a set of m images as input and outputs a set of m region segmentations, one corresponding to each input image. Any region segmentation technique is trivially extendible to a region tracking algorithm simply by applying the technique separately to each image of the sequence, but often it is possible to exploit similarities between consecutive frames of a sequence.

4.2 Thresholding

A basic thresholding technique proceeds by comparing the value at each pixel of an image to a predetermined threshold value. If the value at a pixel is higher than the threshold value, that pixel is classified one way, e.g., as part of the object, and if it is lower, it is classified the other way. Thus the output of such a thresholding procedure is a binary region segmentation. Multiple thresholds can be defined in order to divide an image into more than two separate regions, for example if two thresholds are used, a doubt category can be assigned. For single band grey level images, raw grey level data is often thresholded directly. However, this is generally only effective when images are obtained in such a way as to ensure that it is, for example, objects are silhouetted against a white background. For more complex imagery it is generally necessary to find a function of the pixel data that assigns high values to one class of pixel and low values to the other and apply the threshold to this function of the pixel data.

For multiband data some function must be applied to the data to obtain a 1D measurement that can be thresholded, as, for example, in the works of Ohlander, et al, [OPR78], and Ohta, et al, [OKS80], described in section 2.3.1. This function can take any form but should be selected in such a way as to separate the different classes of data. In particular, statistical measures can readily be used and so it is possible to exploit statistical models in thresholding techniques, a thresholding algorithm that exploits the SCMs of Chapter 3 is described in section 5.1.

Much work has been done on automatic selection of appropriate threshold levels and Sahoo, et al, [SSW88], provide a comprehensive survey of automatic grey-level thresholding
Chapter 4 Region Segmentation and Tracking

techniques; further review of the subject can be found in Haralick and Shapiro, [HS85], and
Zhang, [Zha96]. Lee, et al, [LCP90], Leung and Lam, [LL96], provide comparisons of the
performance of some of the more popular algorithms and threshold selection techniques. Most
of these techniques involve somehow evaluating, or estimating, the performance of the
algorithm at different threshold levels. Performance measures of this type are discussed further
in Chapter 6.

These simple algorithms do not assume or exploit any connectedness or compactness of
the region of interest. While in a sense this makes their general applicability very wide, when
the region of interest is connected and compact many needless misclassifications can occur well
away from the true boundary of the region of interest. Figure 21, in section 5.1 (page 105),
shows binary region segmentations that are typical of those obtained from simple thresholding
procedures. This problem can be ameliorated to some extent by post-processing such as the use
of mathematical morphology, [HSZ87], as shown in Figure 22 in the same section.

4.3 Region Growing

Region growing is a well-established technique for extracting regions of interest from images.
Region growing techniques can be used to extract a single region from an image, as, for
example, in the work of Sivewright, et al, [SKD93], or can be used to solve the full
segmentation problem, as in the algorithms of Klinker, et al, [KSK90], and Bajcsy, et al,
[BLL96]. When a single region is to be extracted from an image, a single seed is defined within
the region of interest as a starting point. The seed is used to characterise the region to be
extracted and may consist of a single pixel or a group of pixels, depending on the complexity of
the region. In some applications, for example computer drawing packages, where the algorithm
is used to fill homogeneous regions outlined by hand, the region can be characterised from a
single sample pixel. In real images, however, regions are rarely homogeneous. In some
applications, a real image is pre-processed, by an edge detector for example, so that it is divided
into homogeneous regions, which can be extracted in the same way. A more common approach,
however, is to extract the statistics of the region from a sample distribution contained in a seed
region. The seed also defines an initial region from which the entire region is extracted by
iteratively adding neighbouring points if they pass some test of similarity to the pixels already
contained in the region. These similarity tests are often based on statistical measures such as
those provided by the statistical models in Chapter 3.

Region growing procedures are more complex when the goal is to fully segment an
image as criteria must be defined for splitting regions in two or joining adjacent regions.
Zucker, [Zuc76], provides a review of region growing techniques, particularly for application to
the general segmentation problem; further review of the subject can be found in [HS85].
In a region segmentation algorithm based on region growing, the connectivity of the region of interest is exploited to a much greater degree than in thresholding based techniques. However, the topology of the extracted region is still not fixed and it may be disconnected and contain holes. Figure 23, in section 5.2 (page 107), shows typical region segmentations obtained from such a process. As for thresholding algorithms, if necessary, post-processing can ameliorate these problems and the effects of a morphological operation on the region segmentations shown in Figure 23 can be seen in Figure 24.

These procedures are readily extendible to data of higher than two dimensions. Sivewright, et al, for example, used a 3D region growing algorithm on CT image data to extract 3D regions corresponding to parts of the brain. In a similar way, region growing techniques can be applied to region tracking tasks and a procedure that exploits a statistical models, such as those described in Chapter 3, for such tasks is described in section 5.2.

4.4 Active Contour Models

In recent years, active contour models (ACMs) have emerged as an effective mechanism for segmentation and tracking of object boundaries in images or image sequences. Typically the contour is defined by a finite number of points in the image, \( u(s) \), where \( s = 1 \ldots n \), and these points are joined to make the contour. If the contour is closed, the first and last points are also joined together. Central to the use of most ACMs is the minimisation of a function that describes the energy of the contour in its current configuration. This energy function typically has two components, an \textit{internal energy} component that applies shape constraints to the model, and an \textit{image energy} derived from the data to which the algorithm is being applied. In theory, the total energy of the contour is the integral of the sum of these energy terms around the contour. In practice, when the contour is described by a finite number of elements, the energy is evaluated at each element and these contributions are summed to yield the total. A well selected energy function will have a clearly defined minimum point when the contour lies along the desired boundary in the image. A good energy minimisation technique is also required to ensure that this energy minimum is reliably found. Since the original formulation, due to Kass, et al, [KWT87], many alternatives have been proposed and this section provides a review of the use and implementation of active contour models.

4.4.1 Original Formulation

In the original formulation, due to Kass, et al, [KWT87], the energy function in fact contains three terms for each element:

\[
E_{\text{contour}} = \sum_{s=1}^{n} \left( E_{\text{internal}}(s) + E_{\text{external}}(s) + E_{\text{image}}(s) \right). \tag{48}
\]
Kass, et al's. active contour was designed for use with a human overseer and the additional external energy is a term that allows points of attraction and repulsion or other user defined interactions to be specified. For a repulsion point for example, the closer a contour element moves to that point, the higher its external energy term. Specification of these points allows the user to manually deter the contour from converging to local energy minima and ensure that the correct image boundary is found. In automatic applications, however, external energy terms are generally not used.

The internal energy term consists of energy terms that strive to minimise stretching and bending of the contour. In [KWT87], the internal energy at a particular contour element consists of two terms:

\[
E_{\text{internal}}(s) = \frac{\alpha}{2} \left( \frac{\partial u}{\partial s}(s) \right)^2 + \frac{\beta}{2} \left( \frac{\partial^2 u}{\partial s^2}(s) \right)^2.
\]  

(49)

The first term in (49) describes the tension energy of the contour. It decreases as the distance between consecutive elements of the contour decreases and so resists stretching of the contour and keeps the overall contour length to a minimum. The second term is the stiffness energy, which serves to minimise bending of the contour, thus keeping it reasonably smooth. \(\alpha\) and \(\beta\) are weighting coefficients of these two energy terms. The final energy term is the image energy and this is defined in terms of whatever kind of image feature the contour is required to find. For example, if

\[
E_{\text{image}}(s) = -\gamma \left( \frac{\partial}{\partial u} G_\sigma * I(u(s)) \right)^2,
\]  

(50)

i.e., minus the derivative of a Gaussian convolution with an image neighbourhood local to the contour element, then the image energy is minimised at regions of high contrast - edges. \(\gamma\) is the weighting coefficient for this energy term. There are many other features that can be used, e.g., lines, high intensity, etc, simply by changing the filter applied to the image in (50), and these energies can even be combined in a single image energy term, see [KWT87, Ivi96]. If there are no external energy terms, the total energy of the contour is given by

\[
E(u) = \sum_{s=1}^{n} \left( \frac{\alpha}{2} \left( \frac{\partial u}{\partial s}(s) \right)^2 + \frac{\beta}{2} \left( \frac{\partial^2 u}{\partial s^2}(s) \right)^2 + \gamma P(u(s)) \right)
\]  

(51)

where \(P(u(s))\) is the "potential energy" equal to the image energy term, for example, the edge potential given in (50). Kass, et al, apply the Euler-Lagrange equations to this energy function to show that extrema occur when

\[
\frac{\partial E}{\partial u} - \gamma \frac{\partial P}{\partial u} - \frac{\alpha}{2} \frac{\partial^2 u}{\partial s^2} + \frac{\beta}{2} \frac{\partial^4 u}{\partial s^4} = 0.
\]  

(52)
at every point around the contour. Such a contour configuration, \( u \), cannot be found analytically and, rather, gradient descent is used to iteratively minimise the energy. Equation (52) suggests that a suitable equation of motion for minimisation of the energy function, (51), by iterative gradient descent is given by,

\[
\frac{\partial u}{\partial t} = -\frac{\partial E}{\partial u} = \alpha \frac{\partial^2 u}{\partial s^2} - \beta \frac{\partial^4 u}{\partial s^4} - \gamma \frac{\partial P}{\partial u}.
\]  

(53)

These discrete derivatives can be approximated using finite differences between consecutive elements of the contour:

\[
\frac{\partial u}{\partial s} (s) = u(s+1) - u(s)
\]

(54)

\[
\frac{\partial^2 u}{\partial s^2} (s) = u(s-1) - 2u(s) + u(s+1)
\]

(55)

\[
\frac{\partial^4 u}{\partial s^4} (s) = u(s-2) - 4u(s-1) + 6u(s) - 4u(s+1) + u(s+2).
\]

(56)

\[
\frac{\partial P}{\partial u} = \left( \frac{\partial P}{\partial x}, \frac{\partial P}{\partial y} \right) = (P(x+\delta x, y) - P(x, y), P(x, y + \delta y) - P(x, y))
\]

(57)

Kass, et al, suggest two possible ways to implement the gradient descent energy minimisation process. In both approaches, a force is applied to each element, \( s \), which is found by evaluating the right hand side of (53), at that element and scaling by a suitable timestep, \( \delta t \), to obtain the appropriate \( \delta u \) - the change in contour element position. The first approach is an explicit Euler timestepping approach where the force is applied to each contour element in turn. This scheme is explicit in the sense that it uses current element positions to guess potentially better positions for the next iteration. The position of each element is updated before moving on to calculate forces on subsequent elements of the contour. In the second approach a new position for each element is found by solving a set of simultaneous equations and the positions of all the elements are updated together. The equations are defined using the new positions in the internal energy terms but current positions for the image energy. The scheme is therefore implicit with respect to the internal energy but explicit with respect to the image energy and is referred to as “semi-implicit”. An nxn matrix has to be inverted in order to solve the simultaneous equations in the semi-implicit approach, but, provided the coefficients of the energy terms and the number of contour elements remains the same, it only has to be done once and so computation times are not significantly affected. Moreover, this matrix is pentadiagonal and so can be inverted very efficiently, see [KWT87]. Both of these schemes have been well documented in the literature, see for example, [KWT87, Ivi96], and so full details are not presented here.
4.4.2 Common Drawbacks

Since the original formulation of the ACM, applications have been wide and varied. Analysis of ACMs in practical situations, however, has revealed some common drawbacks and many reformulations have been devised in an effort to address these problems. Common problems that have been identified with the original formulation include,

- **Sensitivity to initial configuration.** The final position of the contour has often been reported, e.g., [Ber91, Coh91], to be highly dependent on the initial configuration, which must therefore be close to the required boundary to obtain good results.

- **Local minima.** This is related to the previous problem; it has been found that active contours can latch on to spurious weak features in the image and never find stronger features that correspond to the global minimum energy configuration. When the boundary extraction is guided by the user, local minima can be avoided by specifying attraction and repulsion points, but in a fully automatic system this is a serious problem.

- **Non-convergence.** Convergence cannot be guaranteed for active contour models implemented in the way described above and they have a tendency to oscillate about the position of minimum energy. This can make it hard to detect the point at which the contour has reached an image boundary and reduce its final accuracy.

- **Parameter values.** There are no guidelines for selecting values for the coefficients of the different energy terms in the energy function, (51). Different settings for these parameters can cause significantly different behaviour of the model. If the coefficients of the internal energy terms are set too high the contour does not extract fine detail in the true boundary and tends to cut corners, but if they are set too low, the contour is susceptible to noise and may become unnecessarily jagged.

- **Bunching.** It has often been reported, e.g., [Ber91], that rather than spacing evenly along the contour, when there is a particularly strong feature in the vicinity of the contour, many contour elements are attracted towards it causing a bunching effect of elements in the corresponding part of the contour.

- **Shrinking.** Because of the tension energy term in (49), active contour models always strive to decrease their length and in the absence of image forces will shrink to a point under the tension forces. This tendency to shrink can cause problems when the required boundary is outside the initial contour.

4.4.3 Variations of the Energy Function

Many variations and improvements have been suggested both in the form of the energy function itself and the techniques used for minimisation. In this section, suggested alterations to the
energy function are reviewed and alternative energy minimisation schemes are discussed in the next section.

Cohen, [Coh91], introduced a pressure force allowing the contour to expand through weak edges and only be stopped by strong ones. The pressure force acts with fixed magnitude along the outward normal to the contour, which must now be closed, at each element. Ivins, [Ivi96], showed that addition of this force has the effect of maximising the area inside the contour with respect to the other constraints. In fact, it corresponds to the inclusion into the original energy function, (51), of an additional energy term, $E_{\text{pressure}}$, that is a negative scalar multiple of the area inside the contour, $R$:

$$E_{\text{pressure}}(u) = -\rho \int_R dx dy$$

The elemental equation of motion becomes:

$$\frac{\partial u}{\partial t}(s) = \alpha \frac{\partial^2 u}{\partial s^2}(s) - \beta \frac{\partial^4 u}{\partial s^4}(s) + \rho \left( \frac{\partial u}{\partial s}(s) \right)^\perp - \gamma \frac{\partial p}{\partial u}(s).$$

where $\rho$ is the weighting coefficient of the pressure force. The normal is calculated by using finite differences to approximate the first derivative, as in (54), and taking a vector perpendicular to this tangent direction. Cohen’s pressure force allows the contour to expand even when there is no potential gradient pulling it outward and also allows the contour to pass over weak edges enabling it to reach more distant but stronger edges representing deeper minima in the energy surface. The shrinking problem is also ameliorated and the equilibrium position in the absence of image forces is now a circle whose radius depends on the ratio of the tension and pressure coefficients, [Ivi96]. In [Coh91], a few examples are shown in which the original ACM fails to reach the desired boundary from a specified initial configuration, but is successful when the pressure force is incorporated.

This pressure force can introduce its own problems, however. If the desired edge actually resides inside the contour, this force can push the contour away from it. Also, if part of the true boundary has weak edge properties, the pressure force can push the contour over that part of the boundary even if there is no stronger edge lying beyond it. Cohen and Cohen, [CC93], proposed a distance potential force in which a potential field is constructed using a distance transform from a binary edge map. Using this potential field, the contour can detect features from further away and weak features will only be passed over if there is a better alternative in the vicinity. Cohen and Cohen also suggested an alternative approach to the estimation of the discrete derivatives that must be evaluated in the elemental equation of motion. They found that overall stability, in terms of the number of iterations required before a termination condition is met, was increased if these derivatives were estimated using a finite element method rather than finite differences.
Problems still remain when trying to extract boundaries with concavities, however. As part of a contour tries to move into a U-shaped concavity on a boundary, the closest edges will often be those at the sides making it difficult for elements to reach the end. Xu and Prince, [XP97], proposed a new image force based on a flow field computed from an edge map. It retains the nice features of the distance based image force, i.e., allowing boundaries to be found from some distance from either side of the object boundary, and copes well with concavities in the object boundary. The method is shown to work well on edge maps constructed to illustrate this problem.

Techniques such as Cohen’s pressure force and the more sophisticated potential field used by Xu and Prince can reduce the sensitivity to the initial configuration and local minima in the energy surface. A different way to ameliorate this problem is suggested by Leymarie and Levine, [LL93], who successfully used ACMs to track the boundaries of cells through sequences of microscopic images. They suggested the use of a scale space to combat the sensitivity to initial configuration of the models. Initially, the image is blurred with a wide Gaussian filter so that strong edges can be detected from far away. As the number of iterations increases and the ACM approaches the boundary, the image is sharpened to allow accurate localisation of the boundary. In a similar way, Leroy, et al, [LHC96], discuss how multi-resolution imagery might be used to avoid convergence to local minima. Leymarie and Levine also propose a termination condition by thresholding on the value of the ratio of the overall contour energy, (51), to the total contour length. As has become common in the application of ACMs to tracking applications, it is assumed that the difference between consecutive frames of the sequence is small so that the final configuration on one frame is used as a starting point on the next.

Berger, [Ber91], uses a comparison of the edge strength at each element of the contour with the average edge strength over all the elements to obtain a measure of quality of the current contour position. This measure of quality is used to determine when the contour has reached the boundary of interest and thus provides a termination condition independent of numerical instabilities that would otherwise cause the model not to converge. Furthermore the quality measure can be taken on subsections of the contour and thus can provide local quality measures, which can be used to set the values of the coefficients of the energy terms adaptively, thereby alleviating the problem of corner cutting. In [Ber91], an alternative tension energy term is proposed so that, rather than pulling contour elements as close together as possible, its influence is to make inter-element distances as close as possible to some predetermined value. This alleviates problems of elements bunching around strong edge sections as well as the shrinking problem. Some salient contours in real images are shown to be extracted reasonably using an ACM with the proposed enhancements, but no quantitative results or empirical comparison with the original implementation is provided. Eviatar and Somorjai, [ES96], use a similar tension
energy term to that of Berger and further omit the stiffness energy term from the internal energy expression. Omission of the stiffness energy is quite common, particularly when the required boundaries are complex and smoothing the contour could decrease the accuracy of the boundary representation. An example is the work of Davatzikos and Prince, [DP95], who, like Eviatar and Somorjai, use ACMs to extract cortical boundaries from medical imagery. However, no quantitative results are provided in either case to justify this omission.

Fua and Leclerc, [FL90], presented one of the first non-interactive applications of ACMs, in which ACMs were used to extract buildings and roads from aerial photographs. A method for adaptively setting the value of the step size, $\delta t$, is proposed. The change in energy that occurs over a full iteration is checked and if the energy increased, the step size is reduced for subsequent iterations. A termination condition is applied by placing a minimum threshold on the step size. Contours are initialised on edge segments in an edge map, generated using a classical edge detection algorithm, [Can85], which fit a geometric model of salient boundaries in the image. Results on several aerial images are presented in which the final ACM positions appear to be improvements over the initial boundary estimates provided by the edge detector.

The stiffness energy term used by Kass, et al, is designed to be a property of the curvature of the contour. Williams and Shah, [WS92], showed that the approximation to the curvature provided by the finite difference method is not consistent with mathematical definitions of curvature. They proposed some other curvature estimates and compared the consistency of each with mathematical definitions; some improvement over the finite difference approximation was found. It is not clear to what extent use of these alternative stiffness energy formulations affects the overall behaviour of the contour and no quantitative analysis at this level is provided.

To assist the location of the global minimum energy configuration, Gunn and Nixon, [GN97], proposed the use of two contours. One is initialised completely within the region of interest and has an energy function designed so that it will expand outward toward the boundary and the other is initialised outside the region and designed to contract inward. A termination condition, which is a threshold on the total movement of the contour, determines when the equilibrium position has been found. Once both contours have found equilibrium, elements of the different contours are identified and each pair is checked to see if the same minimum energy point has been found. If some corresponding elements have found different minima, the contour with larger energy is perturbed until its energy further decreases and is then allowed to find equilibrium once more. The process continues until the same minimum points have been found all around the contour. Results are shown on two images, one real and one synthetic, in which the original ACM converges to a local minimum whilst the dual contour extracts the salient boundary successfully.
4.4.4 Alternative Energy Minimisation Schemes

Both energy minimisation schemes proposed in [KWT87] are based on variational calculus and can be unstable. Samadani, [Sam91], provides a detailed discussion of the convergent and non-convergent properties of variational ACMs and proves that convergence cannot be guaranteed in general. Other techniques for iterative minimisation of energy functions such as (51) are available however and several alternative approaches to minimisation of the ACM energy have been proposed.

4.4.4.1 Dynamic Programming

Dynamic programming is a well-known approach to the solution of optimisation problems and is discussed in most texts on the subject, see for example [Fou84]. Amini, et al, [AWJ90], suggested that it could be generally useful, as an alternative to variational calculus, for solving variational problems in computer vision. As an example, they showed how it could be used to iteratively minimise the energy of an ACM. A neighbourhood, typically the 3x3 square with the pixel containing the element at the centre, is defined around each element of the contour and the minimum energy configuration of each set of three consecutive elements is found. As the actual value of the energy contribution at each element is calculated, rather than calculating forces as in the variational approach, the order of the discrete derivatives that have to be calculated is lower which can significantly reduce noise in the system. The first and second derivatives in (49) are approximated, as usual, using the standard finite difference approximations given in (54) and (55). The expressions for these derivatives involve only the positions of one contour element and its two neighbours and the image energy contribution from an element is a property of the position of that element alone. A move of one element therefore, affects the energy contributions of that element and both immediate neighbours only. Thus, after each iteration, the locally minimum possible energy configuration is guaranteed. However, this does not ensure that the global minimum is eventually found.

Convergence of the energy minimisation is guaranteed but computational complexity is high. For example, if the elemental neighbourhood is of size m, and there are n contour elements, complexity is \(O(nm^3)\) for each iteration. The procedure is, however, parallelisable up to a granularity of \(m^3\) should resources be available. Although the complexity of the variational approach is \(O(n)\) and so appears significantly favourable, some computational load is added by the fact that element positions are sub-pixel and so must be stored as floating point values. When dynamic programming is used, all element positions are integers. The positions of the elements are updated simultaneously after optimal moves have been calculated. In [AWJ90], results of the application of a dynamically programmed ACM to a number of simple images in which the region of interest is distinct from the background are shown and compared to results obtained using a variational ACM on the same images. Values for the coefficients of the energy
Chapter 4 Region Segmentation and Tracking

terms and other internal parameters, such as number of elements, are fixed to the same values and the configurations of the contours are shown after several fixed numbers of iterations on each image. The instability of the variational approach is clearly evident and the results obtained from the dynamically programmed ACM appear to fit the true image boundaries much more closely. It is not clear, however, how the internal parameter values were selected and no quantitative measurements for comparison are provided.

4.4.4.2 Greedy Algorithms

The use of greedy algorithms, originally suggested by Williams and Shah, [WS92], to minimise the energy of an ACM, has become a popular alternative to the variational approach. In a similar way to dynamic programming, a neighbourhood of candidate new positions is defined around each element. In the implementation presented in [WS92], each element in turn is set at each position in its neighbourhood and the energy contribution from that element in each position is calculated. The position that minimises the energy contribution is chosen as the new position of that element and the position is updated before moving on to update subsequent elements of the contour. Although a single move might decrease the energy contribution from a single element, as mentioned in the last section, it also affects the contributions of the two neighbouring elements, which may increase causing the overall contour energy to increase. Williams and Shah's scheme therefore does not guarantee convergence as oscillations can occur (see section 5.3.6.1 in the next chapter).

A simple extension, however, suggested by Lai and Chin, [LC93, Lai94], is to consider the total energy after each move as in Amini, et al's dynamic programming scheme. As discussed in the last subsection, when one element is moved, only the energy contributions from three elements are affected. Minimising the sum of these contributions over the set of candidate positions means that the total energy of the contour over that set of positions for that element is also minimised. A move is then only made if it decreases the overall energy of the contour and convergence is therefore assured.

Another favourable feature of this approach is speed. As with dynamic programming, all the element positions are stored as integers but complexity is now $O(nm)$ and the algorithm is still parallelisable up to a granularity of the elemental neighbourhood size, $m$.

Extracted boundaries, from ACMs using variational, dynamic programming and greedy energy minimisation schemes on a number of simple images, similar to those used by Amini, et al, are shown in [WS92]. By inspection, none of the three approaches produces significantly better or worse results than the others and no quantitative results are given. The greedy ACM, however, is reported to be significantly more stable than the variational one and faster than the dynamic programming model.
4.4.4.3 Other Methods

Eviatar and Somorjai, [ES96], used Powell's method of conjugate directions which searches directly for the closest minimum point of a function, see [PTV92], to minimise their ACM energy function. They claim that this approach, which does guarantee convergence, is better than other minimisation schemes, but do not justify these claims with results of any kind. This minimisation procedure was also tested by Staib and Duncan, [SD92], who compared its use with a greedy gradient descent approach. They found Powell's approach to be significantly slower, but did not compare actual results obtained using the different approaches. It should be noted, however, that Staib and Duncan used a very different representation for their contour in that work, see section 4.4.5.2. Ivins, [Ivi96], also suggested a conjugate gradients method to refine the contour after a fixed number of iterations of variational minimisation have been applied, but implementation and testing were not carried out.

Leroy, et al, use a Newton method find a minimum point in the energy function. Newton methods often exhibit poor convergence, particularly when the function to which it is applied is complex, and can oscillate around true minima or even diverge. This was reflected in the findings of Leroy, et al, who reported unreliable convergence of their ACM.

4.4.5 Alternative Representations

The representation of the contour in the original ACM implementation is polygonal. Each contour element represents a vertex of a polygon and the whole contour is constructed by joining consecutive vertices with straight lines. The simplicity of this representation is appealing and many subsequent implementations have followed suit. However, other contour representation schemes have been proposed and implemented and applied successfully. These alternative representations are briefly reviewed in this section.

4.4.5.1 B-Spline Representations

A common alternative representation for ACMs, originally used by Menet, et al, [MSM91], and Cipolla and Blake, [CB90], is to represent the contour by a B-spline using the contour elements as knot or control points. B-splines can provide a very compact representation for curves usually requiring less control points, than for polygonal representations for comparable accuracy. Hence fewer equations need to be solved and computation times can be reduced. The use of B-splines also allows for better representations of corners in contours. Because B-splines have implicit smoothness, the internal energy terms of the ACM are not required. Omission of the internal energy terms allows for extremely fast ACM implementations and such models have proved effective for real time applications, as for example in the work of Blake, et al, [IB96, and RWB96].
Saint-Marc, et al, [SRM93], showed that representing image contours using B-splines can facilitate the extraction of symmetries in the image and so may be more useful for higher level processing. They also point out that a limitation of the representation is that it struggles to represent contours of low curvature.

Wang, et al, [WEH96] use a multi-stage ACM based on a B-spline representation to find optimal solutions very quickly. Initially the contour energy is coarsely minimised in the same way as the dynamic contours of Cipolla and Blake; a second stage then refines the contour approximation at regions of high curvature, such as corners. A third stage is then applied to fine tune the contour using an optimisation scheme similar to the dynamic programming approach of Amini, et al, [AWJ90], described section 4.4.4.1. This ACM is shown to represent well a complex boundary containing points, corners and linear sections. It is also shown to successfully find two salient contours in real medical images from initial configurations fairly close to the boundary. No explicit comparison with other models is provided, however.

4.4.5.2 Fourier basis functions

Staib and Duncan, [SD92], represent contour shape by a weighted combination of elliptical Fourier basis functions, or "harmonics". An application specific training set is used to generate an a priori probability model of the set of shapes, consisting of a Gaussian model of the range of each weighting function. The probability of the current contour representing the object of interest is generated by combining the a priori model with evidence from the image and this probability is iteratively maximised using both gradient ascent and a conjugate directions method to obtain the final contour. Boundary extraction examples are shown in clean and noisy synthetic data and real medical images, in which salient boundaries are successfully identified and represented. An evaluation of the dependence on the method on image quality and noise is also provided. Although some increase is found, the average Euclidean distance of the contour from a known image boundary was found to be reasonably insensitive to decreasing signal to noise ratio. A disadvantage of the use of Fourier descriptors for representation of shape is that corners can be troublesome and require a large number of harmonics to be represented adequately.

4.4.5.3 Front Propagation Models

A very different type of representation for ACMs is as propagating fronts. Malladi, et al, [MSV95], proposed a representation of ACMs based on the level sets algorithm for modelling propagating solid-liquid interfaces and flame fronts proposed by Osher and Sethian, [OS88]. These models can change topology during their evolution, i.e., they can split and merge producing islands and holes in the extracted region. One drawback is that corners and cusps in the desired contour may not be well represented. A similar idea was independently proposed by
Chapter 4 Region Segmentation and Tracking

Caselles, et al, [CCC93], on which later work, [CKS95], on geodesic active contours was based by combining these models with energy based formulations. Rougon, [Rou93], provides an analysis of the kinematics of ACMs using a similar propagation model.

ACMs of this type have been used successfully to extract salient regions in medical imagery in [MSV95] and in more recent work by Yezzi, et al, [YKK97]. They have also been applied to the general image segmentation problem by using several seed contours in the work of Tek and Kimia, [TK95]. The main advantages of these approaches are the ability to represent shapes with varying topology. However, McInerney and Terzopoulos, [MT95], proposed a method for allowing adaptable topology in the more familiar ACMs based on energy minimisation techniques. For complex shapes, variable topology may be a valuable asset, but it does render contours more susceptible to noise and when regions of interest have simple, constant topology, may even prove disadvantageous. The author knows of no empirical comparison of this type of model with other ACMs.

4.4.6 ACMs and Colour

Most of the ACMs discussed above work on single band grey level data by searching for features such as edges or lines. Any of these ACMs could therefore be applied to multiband imagery simply by changing the definition of the image energy to provide an affinity for multidimensional features, e.g., by using a multidimensional edge filter. Applying the same filter to each band of the image and combining the results in some way might provide some extra information, but often better results are obtained by using the bands together. Sapiro, [Sap96], proposes a general technique for applying ACMs to multiband imagery in which an optimal multiband edge detector is used to drive a geodesic active contour, [CKS95]. No report of experimentation with these models is given, however.

Ngoi and Jia, [NJ96], use a colour based image energy given by

\[ E_{\text{image}} = \gamma E_{\text{contrast}} + (1-\gamma) E_{\text{intensity}} \]  

where \( E_{\text{contrast}} \) and \( E_{\text{intensity}} \) are heuristically defined energy functions describing the chromaticity and intensity differences, respectively, from a reference colour vector. The parameter \( \gamma \) is automatically selected by analysing the variances of the two energy terms over regions known to correspond to object and background. Some examples are shown in which their ACM is shown to have extracted various boundaries in natural scenes. However, it is not clear how the contours were initialised or whether the boundaries extracted are of particular interest.

4.4.7 General Comment

From the work discussed in this section, ACMs appear to be a useful technique for the region tracking tasks of interest in this thesis. Clearly, many variants of the original ACM have been
Chapter 4 Region Segmentation and Tracking

proposed. Most modifications are developed to ameliorate specific problems and in the majority of publications documenting such work, examples are shown on one or two images of results obtained using the modified algorithm. In many cases, these examples are not even compared with similar examples obtained from other, unmodified ACMs. Even in cases where results using various implementations are presented, it is often unclear how internal parameter settings were selected and what factors can affect the performance of the different models and how. Such results are not sufficient in general to make informed decisions about how best to implement an ACM for a particular application. For example, how should the internal energy best be represented? Which energy minimisation technique should be used? Which termination condition should be adopted? As little or no quantitative comparison of the options is available, decisions are often based on intuition or trial and error on a few test images, which may fail to highlight hidden dangers.

None of the ACMs discussed in this section are designed to exploit an underlying statistical model and so none of them are directly applicable to the goals of this thesis. In the next section, however, a strongly related technique is discussed that is based around statistical models. Much of the theory that has been developed for ACMs is equally applicable for these active region models and many of the implementational options still remain. Decisions concerning these options that have been made in this work are discussed in section 5.3.

4.5 Active Region Models

Although many modifications and improvements have been suggested, sensitivity to the initial configuration remains a serious problem for ACMs and, especially for complex real world images, they are only really effective when the initial estimate of the true boundary is good. Region growing techniques, section 4.3, have the advantage that the boundary of the region of interest does not need to be detected directly from the initial region specified in the image as a starting point for the algorithm. So long as a good model of the data comprising the region as a whole is available, the region can be grown over arbitrarily large distances to find the region boundary. The disadvantage of region growing strategies is the lack of built-in constraints on connectivity of the extracted region and smoothness of its boundary. The resultant sensitivity to noise causes the boundary of the extracted region to be highly irregular and regions extracted often include holes. Techniques such as mathematical morphology can help, but at the cost of further processing which introduces its own artefacts and increases computation times.

Recently, several algorithms have been developed that combine the techniques of ACMs and region growing. The ACM image energy term is replaced by a term derived from local region information. Points on the contour are allowed to expand or contract according to the fit between local region information and a model of the data comprising the region of interest. The resulting active region models (ARMs) retain the desirable characteristics of both
Chapter 4 Region Segmentation and Tracking

de ACMs and region growing. Regularity of the contour can be controlled by the shape constraints in the energy function. Also, by examining local region information, boundary points are able to traverse large homogeneous areas of the image, to find a distant part of the true boundary, reducing dependence on the initial configuration.

4.5.1 The statistical snake

Ivins and Porrill’s “statistical snake”, [IP94, Ivi96] is based, to a large extent, on the original ACM implementation described in [KWT87]. An energy function, $E$, is specified:

$$E = \frac{\alpha}{2} \sum \left| \frac{\partial u}{\partial s} (s) \right|^2 + \frac{\beta}{2} \sum \left| \frac{\partial^2 u}{\partial s^2} (s) \right|^2 - \rho \int_{R} G(I(x,y)) dx dy .$$

(61)

The first two terms in (61) correspond respectively to the tension and stiffness energies and together comprise the internal energy, exactly as in (51). The third term is the image energy. $G$ is the goodness function derived from a statistical model of the underlying data, exactly as described by equation (17), in section 3.1. $R$ is the region contained within the contour, and $\alpha$, $\beta$, and $\rho$ are the coefficients of the three energy terms. The algorithm is initially given a polygonal region of the image known to belong to the object of interest. This seed region serves two purposes. It is used as the initial configuration of the model – the vertices of the polygonal boundary become contour elements, and to provide a sample distribution from which to construct a statistical model of the data comprising the region as a whole, as described in Chapter 3.

As may be seen in (61), the image energy is minus the integral of the goodness over the interior, $R$, of the ARM, scaled by the coefficient $\rho$. The goodness is defined, in (17), in such a way that pixels with similar values to the sample in the seed region are assigned positive goodness and, as the values become more different, the goodness decreases eventually tending to minus infinity. The image energy is the sum of the goodness of every pixel within the contour and is thus minimised by including as many pixels with positive goodness as possible, while excluding those with negative goodness. This image energy term is similar to the pressure energy term, $E_{\text{pressure}}$ in equation (58), and if the goodness function is identically one, the two energy terms are equivalent. The force derived from the image energy is thus defined, in a similar way to Cohen’s pressure force, to act along the normal to the contour with magnitude proportional to a local goodness measure. The overall force on an element is then given by the following elemental equation of motion:

$$\frac{\partial u}{\partial t} (s) = \alpha \left( \frac{\partial^2 u}{\partial s^2} (s) \right) - \beta \left( \frac{\partial^3 u}{\partial s^3} (s) \right) + \rho G(I(u(s))) \left( \frac{\partial u(s)}{\partial s} \right)^4 ,$$

(62)

and the displacement is found in the usual way, by multiplying the right hand side of (62) by a timestep $\delta t$. 

96
In order to avoid sensitivity to noise, the goodness of a local neighbourhood average pixel value is used. Ivins and Porrill take the average pixel value over the set comprising the two lines joining an element to each of its neighbours, which are found using a Bresenham line drawing algorithm, [FVF90]. The goodness of this average pixel value is used as the local goodness measure \( G(I(u(s))) \), in (62). In addition, following Cohen, [Coh91], to aid stability of the model and to simplify the avoidance of self-intersection discussed in the next paragraph, each displacement, \( 5u(s) \) is normalised to unity if its magnitude is greater than one. Ivins and Porrill use the explicit variational approach to minimise the energy. This is preferred to the semi-implicit method as it facilitates the implementation of the following two constraints on the model.

4.5.1.1 Avoiding Self-Intersection

For the ARM introduced above, self-intersection is a serious problem. If self-intersection should occur, a portion of the contour effectively becomes inside-out. The image force is thus reversed causing contraction forces to become expansion forces and vice versa, see [Ivi96]. Typically this occurs when a portion of the contour is in a particularly undesirable region of the image and is contracting very rapidly (see Figure 27 on page 111). Thus, once self-intersection has occurred, the portion of the boundary where it has occurred is likely to be surrounded by undesirable pixels. This causes strong contraction forces to be applied to that portion of the boundary, but as these forces now act outward, uncontrollable expansion occurs.

Self-intersection is detected by keeping a 2D-accumulator array of positions in the image that are already occupied by the contour into which other sections of the contour cannot move. Such positions include not only the actual positions of any other elements, but also repulsion zones about them and lines joining consecutive elements. Thus, the 3x3 pixel square centred on each other element is marked as occupied together with 4-way connected lines joining each pair of consecutive elements not including the one currently being updated. If one or more pixels in the repulsion zone of a newly calculated position, or any pixel along the lines joining the new position to the neighbouring elements, is occupied, the move is reversed, i.e., the displacement vector found from (62), is negated. If this reversed move also causes an intersection, the move is cancelled and the element remains where it is. This guarantees that self-intersection cannot occur. Full implementational details can be found in [Ivi96].

4.5.1.2 Addition and Deletion of Elements

Whilst deforming to fit some image boundary, some portions of the contour may have to become significantly longer and other parts may become much shorter. In a section of the contour that is expanding, elements will tend to become further and further apart. This has two effects, firstly, once the boundary is reached, the resolution with which it can be represented
Chapter 4 Region Segmentation and Tracking

will be low and so approximation is poor. Secondly, as the elements move apart, the tension force increases and can eventually outweigh the image force and stop the expanding contour before the true boundary is reached. This situation is resolved by adding new elements to the contour whenever a threshold distance between two consecutive elements is exceeded.

Conversely, as a portion of the boundary contracts, elements can become bunched together. Again there are two significant effects. First, there are far more elements than are required to represent the corresponding portion of the true boundary, needlessly increasing computation. Secondly, as elements become very close, their repulsion zones may begin to overlap and the collision resolution scheme discussed in the previous subsection will not allow them to reach the object boundary. This situation is similarly resolved by setting a lower threshold on the distance between consecutive elements, below which, one of the elements is removed from the contour. Care must be taken when adding elements to or removing elements from the boundary, that no self-intersection is caused.

This scheme is simple to implement when employing an explicit update mechanism and there is little computational overhead. However, when a semi-implicit scheme is adopted, such a scheme can introduce significant extra processing, see [Ivi96]. Addition and deletion of elements, as described above, effects the overall energy balance of the contour, see [Ivi96]. However, this is unimportant in most cases, as the overall energy of the contour is not required.

In [IP94, IP94a and IP95], several examples of the application of the statistical snake to both real medical multiband image data and textured images, which have been filtered with various kernels to produce multiband data, are shown. The algorithm is shown to extract complex boundaries accurately even when the seed region boundary is some distance from the true image boundary. In [Ivi96], further experiments are detailed in which the technique is applied to tracking tasks. Ivins reports that a distinctly coloured card, attached to a moving robot arm, can be tracked accurately through a sequence of colour images, taken in real-time from a video camera via a frame grabber, for several hours without failure.

4.5.2 The Anticipating Snake

Ronfard, [Ron94], independently developed another ARM, the “anticipating snake”. Again, a statistical model of the region is generated and a fit of a local neighbourhood to that model is used to generate a force at a point of the contour. The model is based on a B-spline representation, similar to the ACMs described in section 4.4.5.1. An energy minimisation scheme similar to the greedy approach, discussed in section 4.4.4.2, is adopted, with the set of candidate new positions lying along lines perpendicular to the boundary at each contour element. Resampling of the contour at regular intervals, after every iteration satisfies the need for addition and deletion of contour elements.
Ronfard introduces many heuristics into the algorithm in an attempt to improve stability and overall performance and eventually obtains results that are reported to be comparable to ACMs, but no direct comparisons are presented. Results of the application of the anticipating snake to several tasks involving grey-scale images are shown in which extracted contours appear to be reasonably accurate.

4.5.3 Region Competition

As discussed to some extent in section 2.3.1, Zhu and Yuille, [ZY96], proposed an algorithm, called "region competition", for full segmentation of an image into homogeneous regions. Each region in the image is defined by a contour, which is grown from a seed region using a statistical model and internal shape constraints to calculate forces at points on a contour representing the boundary of that region. When regions meet, edges are formed and points on these edges belong to the boundaries of more than one region. To find forces applied to these points, the forces are calculated from each contour on which the point lies and summed to give the overall force. The internal force on a contour element is designed to minimise the length of the contour and is similar to the standard ACM internal force with the stiffness coefficient set to zero. The image force is similar to that used in the statistical snake of section 4.5.1. However, the log likelihood rather than the goodness measure of (17) (i.e., the goodness with k set to 1), of pixel values is used and is averaged over a circular region centred on a contour element before being used to scale a unit force vector normal to the contour. After convergence is detected, a second stage merges adjacent regions if the overall energy summed over all the contours is decreased. If regions were merged, all the statistical models are recomputed and the individual contours are optimised again, otherwise the algorithm terminates.

4.5.4 Other ARMs

Chakraborty, et al, [CSD96], use the Fourier contour representation method of Staib and Duncan, [SD92], described in section 4.4.5.2, to implement an ARM which incorporates prior shape knowledge, edge information and region information. Initially an edge map is calculated and a "region classified image", based on the use of a Markov random field, is constructed. Pixels in the region classified image are labelled 1 if they are likely to be part of the region of interest and -1 otherwise. An energy term is constructed from the prior model by examining similarity of the weightings of the Fourier basis functions to those obtained over a training set of shapes, as in [SD92]. An edge energy is found by evaluating the integral of the edge map around the contour and a region energy is defined as the integral of the binary region classified image within the contour. The energy is minimised using a conjugate gradients method. In order to calculate the energy gradient efficiently, the region energy area integral is converted to a line integral, in much the same way as the statistical snake, but using Green's theorem, [Kre93], for
justification. Application of the algorithm to several grey-scale medical images shows efficacy of the approach, and the robustness to occlusion that is obtained by incorporation of shape knowledge derived from a training set is demonstrated on a synthetic image. Insensitivity to initial configuration is also demonstrated by measuring the variance of the final configuration of the contour from several different starting points, although the range of variation of these starting points is not shown. That variance is shown to be around half that of contours carefully placed on the image by domain experts. However, no absolute measures of performance are generated.

Figueiredo, et al, [FLJ97], use a two stage minimisation procedure similar to that of Zhu and Yuille's region competition algorithm but where there is only one region of interest. Parameters of statistical models of that region and the background are computed during one stage and a B-spline contour with image energy derived from the statistical models is optimised during the other. Preliminary results show successful application of the algorithm to the task of extracting boundaries in several grey-scale synthetic and real medical images.

4.6 Shape Constraints

ACMs and ARMs can encode a certain amount of knowledge about the shape of the boundary of the region they are to extract by virtue of the weighting parameters, i.e., the coefficients of the different terms, in the energy function. When required boundaries are known to be very smooth, high settings of the tension and stiffness parameters ensure that the extracted boundary will be smooth, but when the boundary is complex, possibly containing corners, these values, particularly the stiffness coefficient, are set low. Moreover, the parameter settings at different contour elements can be set differently if knowledge of the structure of the boundary is known and Olstad and Torp, [OT96], proposed a grammatical framework for encoding shape knowledge into polygonal ACMs in this way. However, in general, it is difficult to determine the optimum parameter settings for representing particular contour shapes and it is not clear how individual contour elements with particular parameter settings are identified with particular points on the true boundary.

When a more global representation of the contour is used, like the Fourier description used by Staib and Duncan, [SD92], as discussed in section 4.4.5.2, shape knowledge from a training set can be incorporated as a prior distribution on the weights of the Fourier basis functions. Incorporation of shape knowledge can decrease the failure rate of a tracking algorithm and also provide some robustness to occlusion of the region of interest. The Point Distribution Models (PDMs) of Cootes and Taylor, [CT92], provide a similar framework for polygonal contours. Landmark points are identified by hand over a training set of images containing instances of some class of object of interest. The landmark points are written as a vector and a principle components analysis identifies the mean and most significant modes of
variation of the overall shape. PDMs can be used to extract new instances of the region of interest by using the mean shape as a starting point and deforming the model, according to the main modes of variation, so that it fits strong local edges found in the image. PDMs have proved very effective when the variation in shape of the region of interest is sufficiently restricted to be represented by a training set and applications have ranged from tracking animals in scenes, [MO95, MT97], to modelling of faces, [LTC94], and the location of resistors on circuit boards, [CT92].

The deformable templates of Yuille and Hallinan, [YH92], provide a similar way to encode shape information into an active contour model. Rather than being defined by landmark point, however, these templates are defined by a number of parametric curves and from some initial configuration, the parameters of the curves are adjusted so that the curves fit features in the image. Adjustment of the parameters is an energy minimisation process similar to that which deforms standard ACMs and in Yuille and Hallinan’s work, the image energy is designed to attract the curves of the template towards edges in the image. Recently, Deng and Lai, [DL97], proposed a region based deformation strategy for deformable templates based on the work of Ronfard, [Ron94], see section 4.5.2. Their algorithm is used to track eyes through sequences of images of human faces, and performance is reported to be reliable.

4.7 Dynamic Models

Motion and change of shape of object boundaries through sequences of images is often highly predictable – a change in position from one frame to the next is usually followed by a very similar change in position in the subsequent frame. Rather than simply using the final position on one frame as a starting point for the next, as is common when using algorithms such as ACMs for tracking, it may be possible to predict the position of the contour in the next frame by analysing the motion over previous frames. Such a prediction scheme could both decrease computation times, as fewer iterations of the energy minimisation procedure are required, and increase robustness of the algorithm as the true contour is more likely to be found from a closer starting point. Terzopoulos and Szeliski, [TS92], used a Kalman filter to predict both the position and velocity (with respect to the 2D image grid) of each element of an ACM and reported that reliability of the tracking algorithm was increased.

Ivins and Porrill, [Ivi96], used an affine field to predict the motion of the region of interest between consecutive frames. In a tracking application, two stages to the energy minimisation of the statistical snake are used. In the first stage, deformation of the contour is restricted to affine transformations only and, once convergence is detected, a second stage refines the contour in the usual way. Enhanced performance, in terms of both speed and reliability, is reported when this scheme is adopted.
Chapter 4 Region Segmentation and Tracking

Recently, a more sophisticated probabilistic model combining shape and dynamics and that can change over time has been proposed by Isard and Blake, [IB96]. Real world testing has shown that active contours incorporating this model are more robust, when applied to tracking objects through cluttered environments, than models incorporating simpler dynamic models such as the Kalman filter.
Chapter 5 Statistical Region Tracking

The previous chapter provides a review of techniques available for tracking of regions of interest, through sequences of images. In this chapter the algorithms that have been used in this work are described. Each algorithm uses a statistical model of the region of interest, such as those discussed in Chapter 3, and is designed so that it can exploit any of these models and hence can be used to compare them. Three different types of algorithm are presented, each of which produces a binary region segmentation for each image of the input sequence. The topology of the typical regions extracted by each algorithm, however, differs considerably. The first algorithm, described in section 5.1, is a simple thresholding procedure, similar to that used by Crisman in [Cri92], but incorporating a morphological post-processing step. The second algorithm, described in section 5.2, is a standard region growing algorithm. The third type of algorithm is the active region model. Using Ivins and Porrill’s active region model, described in section 4.5.1, as a starting point, some problems are identified and some possible solutions are suggested. Several new implementations of the active region model result and these are summarised in Table 3 below, which indicates which implementations are original contributions of this thesis. The abbreviations are described in the text of the sections corresponding to each model, which are also given in Table 3.

<table>
<thead>
<tr>
<th>ARM</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Ivins and Porrill</td>
<td>OIP</td>
<td>The original implementation, as described in section 4.5.1.</td>
</tr>
<tr>
<td>Collision Implies Deletion</td>
<td>CID⁺</td>
<td>As above but with the element deletion, collision resolution strategy of section 5.3.2.</td>
</tr>
<tr>
<td>Discrete Average Goodness</td>
<td>DAG⁺</td>
<td>As CID, but the goodness along the discrete lines joining two elements is averaged rather than the pixel values.</td>
</tr>
<tr>
<td>Single Line Greedy</td>
<td>SLG⁺</td>
<td>The fast greedy implementation of section 5.3.6.1.1 where only a single goodness average is calculated.</td>
</tr>
<tr>
<td>Multi-Line Greedy</td>
<td>MLG⁺</td>
<td>The improved multi-line model given in section 5.3.6.1.2, in which single element two-step oscillations are eliminated.</td>
</tr>
<tr>
<td>Surface Integral Greedy</td>
<td>SIG⁺</td>
<td>Greedy ARM using a surface integral of the bilinearly interpolated goodness function to calculate consistent image energy change approximations. See section 5.3.6.1.3.</td>
</tr>
</tbody>
</table>

Table 3 Summary description of ARM implementations used in this work. * indicates that the ARM is new and so is an original contribution of this thesis.

Although the issue is secondary for the immediate goals of this project, computation times are quoted for all the different algorithms considered in this chapter. These measurements are obtained from a standalone PC, with a P133 processor and 64M RAM, running Linux. Quoted times are not strict, in the sense that they include processor time used to initialise data.
structures and read and write data to and from memory. However, the time required for these operations, is firstly small compared to the time devoted to running the algorithms and secondly reasonably consistent between algorithms particularly as the machine from which they were obtained is not networked. No specialised image handling hardware was used and all the code, which has not been fully optimised, was written in C++ using the standard GNU compiler. The times are quoted for running the algorithm, using a Gaussian SCM with the tolerance parameter, k, set to 3.0, over the image sequence, from which the track image in Figure 3 was taken, which contains 26 frames. A look-up table is used with each SCM so that once a likelihood for a particular colour vector has been calculated, the value is stored and does not have to be computed again. Use of this look-up table ensures that computation times are similar for different SCMs. Quoted times include the time required to construct the statistical model from the sample distribution, which is approximately 2 seconds for the Gaussian model of the track seed region shown in Figure 3. Full discussion of the implementation of SCMs is given in Appendix B, section B.2, together with details of the computation times required to construct the different types of SCM.

5.1 Thresholded Likelihood images

A likelihood image is generated directly from a SCM. For each pixel in an image, the likelihood is found from (18), to yield a likelihood image. Crisman, [Cri92], uses such likelihood images for matching against template binary segmentations in her road following system. Crisman performs a hierarchical search to maximise a match score, p, given by,

$$p = 1 - \frac{1}{RC} \sum_{r=1}^{R} \sum_{c=1}^{C} [p(r,c) - q(r,c)],$$

where R and C are the numbers of rows and columns, respectively, in the image; p is the likelihood image; and q is a binary template image. In this thesis, the likelihood image is scaled to the range [0, 255] yielding a grey scale likelihood image, which is simply thresholded at some grey-level to yield a binary region segmentation. The region tracking procedure based on this simple technique starts with the construction of a statistical model of the region from the seed region given in the first image of the sequence. That model is then used to create a likelihood image from each image in the sequence, which is thresholded to obtain a region segmentation. Figure 20 shows grey scale likelihood images generated from Gaussian statistical colour models of the seed regions in Figure 3 and Figure 21 shows the binary region segmentations produced by thresholding these likelihood images.

This simple algorithm is probably the easiest way to obtain a pixel classification from a SCM, as no assumptions are made about the shape of the region of interest. As a result, performance of the algorithm depends directly on the ability of the SCM to distinguish pixels.
from the parent distribution of the region of interest from background pixels. Likelihood images are reasonably fast to compute, compared to the other algorithms described later in this chapter. If a Gaussian SCM is used, with the tolerance parameter fixed at 3.0, then the processor time required to compute and threshold a likelihood image from each of the 26 images in the track sequence was 6 minutes and 50 seconds.

Many spurious misclassifications well away from the true boundary of the regions of interest can be observed in the region segmentations shown in Figure 21. The regions of interest are both connected and compact and this knowledge can be exploited to improve the results by...
applying mathematical morphology, [HSZ87], to the raw output. In Figure 22 the effects of a morphological closure followed by an opening with a circular structuring element of diameter 9 pixels (found to be optimal in experiments reported later in Chapter 7), on the thresholded likelihood images of Figure 21 are shown. Although overall classification performance appears (see Figure 22) to be enhanced, the inclusion of this post-processing step increases the computation time. Inclusion of the post-processing step increases the processor time required for the track image sequence to 40 minutes and 44 seconds.

In this work, no automatic threshold selection technique is used and performance of the algorithm is characterised independently of the threshold setting, leaving the choice of actual operating point to be selected at a later stage for a particular application, see Chapter 6 and Chapter 7.

5.2 Region Growing

The region growing algorithm used here can utilise any of the statistical models discussed in Chapter 3. As usual the SCM is constructed from the seed region defined in the first image and the tolerance is set at some suitable level. The seed region also serves as a starting point for the region growing process. The algorithm, detailed fully in Appendix B, section B.4, is essentially a depth first search from the boundary of the seed region using positivity of the goodness as a criterion for inclusion of a pixel into the extracted region. Unlike the thresholded likelihood algorithm, the region growing algorithm does not require that the SCM analyse every pixel of every image. There is an implicit assumption of connectedness of the region of interest, as once a boundary of undesirable pixels is found no exploration beyond that boundary is performed. However, this assumption is not strictly enforced and the topology of the extracted region is not fixed – it can contain holes and there can be many distinct connected components. Figure 23 shows region segmentations produced by this algorithm on the images shown in Figure 3. Once again, as the regions of interest are known to be connected and compact, morphological post-processing can improve classification rates by filling in holes in both the positively and negatively classified regions. Figure 24 shows the results of post-processing the region segmentations of Figure 23 by a morphological operation.

When this algorithm is used for tracking, the final configuration on one image is used as the initial configuration on the next. The region segmentation on the previous image must be processed to extract the new list of boundary points from which to start the depth first search. As the algorithm moves down a sequence of images, the boundary of the extracted region can become very complex causing the initial boundary lists on the subsequent frames to be very long. This may cause computation times to be high for subsequent frames of the sequence. Figure 25 shows the region segmentations produced using this strategy on the next four frames.
Figure 23 Regions extracted using the region growing algorithm of this section with a Gaussian statistical colour model with $k$ set to 3.0.

Figure 24 Effects of mathematical morphology on the regions extracted by the region growing algorithm, shown in Figure 23. A morphological closure is performed followed by an opening using a circular structuring element of diameter 9 pixels.

Figure 25 The four subsequent region segmentations for the track sequence following those in Figure 23 (top – no morphology) and Figure 24 (bottom – morphological post-processing). of the tracks sequence with and without the morphological post-processing step, i.e., the region segmentations following those shown in Figure 23 and Figure 24. The smoothing that is achieved by post processing with morphological filters greatly reduces the complexity of the extracted boundary, which otherwise includes the pixels neighbouring each single pixel hole in the region of interest. Thus, despite the extra processing stage, use of the post-processing step can decrease overall computation times. The overall processor time required for the track sequence with no post-processing was 56 minutes and 36 seconds. When each region segmentation was post-processed, as shown in Figure 24 and Figure 25, the processor time was
reduced to 50 minutes and 24 seconds. The effect on computation time of the morphological post-processing is much more noticeable at extreme settings of the tolerance parameter, k.

5.3 Active Region Models

ARMs have proved to be an effective technique for exploiting SCMs for tracking the boundaries of regions of interest through sequences of colour images. In preliminary work, reported in [AB96, AB96a, AB97], a version of Ivins and Porrill's ARM, described in section 4.5.1, was used to compare the performance of various SCMs for region tracking. Although Ivins' original statistical snake code was kindly donated, which helped greatly in the development of many of the ideas in this chapter and the rest of the thesis, the implementation was geared towards a different application and was somewhat inconvenient for the work reported here. The implementation used in both preliminary work and in this thesis is an entirely new version written by the author. No formal test of equivalence of the two implementations has been performed, but the original implementation was available throughout the re-implementation allowing visual comparisons to be made. The two implementations appear to produce similar results when applied to the same imagery with the parameters set to equivalent values.

To measure processor times for the track sequence, a fixed number of 500 iterations of the energy minimisation procedure is set for the first image of the sequence and 200 for subsequent images. The energy coefficients are set to $\alpha=1.0$, $\beta=0.5$, $\rho=5.0$; the timestep, $\delta t$, is set to unity; and the thresholds on the distance between consecutive elements of the contour, above and below which elements are added to or deleted from the contour, are set to 15 and 3 pixels, respectively. These same parameter settings, selection of which is discussed in Chapter 7, are used throughout this section unless it is explicitly stated otherwise. The time required for the new version of the original Ivins and Porrill ARM (henceforth referred to as the OIP ARM) is 16 minutes and 26 seconds. As there are 26 images in the sequence, 5500 iterations are performed and thus a rate of about 6 iterations per second is achieved. It should be noted that, using specialised image handling hardware and ARMs with fewer contour elements, Ivins' original implementation could work at frame rate performing 50 iterations per frame. The use of an ARM algorithm assumes that the region of interest is both connected and fairly compact and usually (but see section 4.4.5.3), as with all the ARMs used in this thesis, the topology of the region is fixed.

Typical region segmentations obtained from an ARM, with the parameters set as described above, are shown in Figure 26. The contour outlining the track contains roughly 200 elements and that for the shorts about 100. The positively classified region is identified using a
version of the raster scan polygon-filling algorithm, in [FVF90], in which only pixels completely inside the contour are classified positively.

Figure 26 Typical region segmentations obtained from an ARM applied to the images of Figure 3.

Ronfard’s anticipating snake, [Ron94], described in section 4.5.2, contains many heuristic solutions to problems encountered during the development of the algorithm. The resulting algorithm is quite complicated, containing many internal parameters, and it is not clear which of the several implementation options should be used in a given application. A restricted version of Zhu and Yuille’s region competition algorithm, [ZY96], described in section 4.5.3, where only two regions are allowed is extremely similar to the statistical snake. The only differences in this case lie in the form of the internal contour energy and the neighbourhood over which local averaging is performed. No detailed quantitative analysis of any of these algorithms has been performed so there is no obvious way to select the most appropriate. However, the simplicity of the statistical snake makes it preferable to the anticipating snake and, although in essence the algorithms are quite similar, the fact that the statistical snake is already geared towards tracking applications makes it preferable to region competition. The use of alternative representations such as Fourier basis functions and B-splines, as in the works of Chakraborty, et al, [CSD96], and Figueiredo, et al, [FLJ97], see section 4.5.4, has not been considered in this work. ARMs based on these representations may prove to be viable alternatives for a tracking system, however, and a comparison would doubtless be useful but is beyond the scope of this project. The main objective here is the comparison of the colour
models and the purpose of this chapter is to find an algorithm whose performance can be characterised reliably and so can provide a trustworthy comparison of the underlying models.

While applying the statistical snake to the tasks of interest here and attempting to characterise its performance some problems with this implementation were identified. These problems are addressed in the remainder of this section and some reformulations of the original algorithm are proposed to ameliorate them. A preliminary report of this work has been published in [AB97a].

5.3.1 Problems

There are two immediately obvious problems with the original algorithm, both of which reduce the accuracy of the final position of the contour.

- Deadlock. Although the self-intersection detection mechanism described in section 4.5.1.1 is very effective and self-intersection never occurs, the collision resolution strategy adopted is not very effective and it is fairly common to find deadlocked portions of the boundary, see Figure 27. When the ARM is used to compare performance of different SCMs, this effect introduces artefacts into the performance measure, for example increasing the number of misclassifications, which can corrupt the final comparison.

- Convergence. ARMs (and ACMs) such as the statistical snake that are implemented using the variational approach to energy minimisation, described in section 4.4.1, do not converge. Once the desired boundary is found, small perpendicular oscillations about that boundary tend to occur as well as a slow global tangential rotation about the contour as a whole. These effects make it difficult to detect the point beyond which no further improvement is made. In practice, [Ivi96, AB97], the number of iterations is set to a high value, but this is computationally wasteful and may hinder overall performance.

Other common problems that occur with ACMs, listed in 4.4.2, do not occur or are less serious in the ARM. Bunching of contour elements around strong features does not occur, as elements are not attracted to features in the same way; for the same reason there is no problem with the extraction of concave portions of an image boundary. The ARM contour will still shrink to a point in the absence of image forces, but this problem is not significant in practice as the image force can act inward or outward with sufficient strength to overcome this effect.

5.3.2 Deadlock

The solution to the problem of deadlock is remarkably simple. When a calculated move causes self-intersection, rather than reversing the move, as was suggested by Ivins and Porrill, see section 4.5.1.1, the offending element is removed from the contour. Obviously, it is possible that the deletion of such an element could itself cause self-intersection. This must be checked
against the 2D accumulator, and, if such a situation arises, the original collision resolution strategy is reverted to, so the move is reversed.

Figure 27 Evolution of two ARMs one (top) using Ivins and Porrill's collision resolution strategy and one (bottom) using element deletion.

This strategy has proved to be extremely effective and both the problem and the solution are illustrated in Figure 27. ARMs using each collision resolution strategy are started from an initial configuration (to the left of the figure) outside the region of interest, which is the green circle. The ARMs use a Gaussian SCM identical to the one from which the colours of the pixels inside the region of interest are drawn. The figure shows the configuration of the contour as the number of iterations increases; from left to right the number of iterations is 0, 25, 50, 100, 200. The upper sequence shows the ARM in which moves are reversed when a collision with another part of the boundary is detected. Part of the boundary becomes tangled and never recovers. The lower sequence shows the ARM in which elements are deleted and it can be seen that the tangle that developed in the other ARM does not occur and a good approximation to the true boundary is eventually found.

Although the example given is purely illustrative and contrived so that the ARM shown in the top sequence will become tangled, such situations are not uncommon in practice. Protrusions in the region of interest in one frame of a sequence can leave the corresponding part of the contour surrounded by background pixels in the next frame. That part of the contour then must contract quickly and may become tangled.

The ARM using this new collision resolution is referred to as the Collision Implies Deletion (CID) ARM in the remainder of this thesis. Often, the behaviour of the CID ARM is identical to the OIP and the processor time for the track sequence with the moderate setting for \( k \) is the same. Differences are more common at more extreme settings of \( k \).

5.3.3 Convergence

The ability to detect the point at which no further improvement to the contour will occur is highly desirable for two reasons. Firstly, if a reasonable time to cease iteration of the energy
minimisation procedure can be detected, knowing that the boundary approximation will not significantly improve, computation time can be greatly reduced. If no such termination condition is employed, the fixed number of iterations used must be set at a level at least comparable to the maximum number of iterations that would be required to find a good approximation to the true boundary for any image in the application domain. Secondly, when a system goes into oscillation about its ideal state, it will pass through many different configurations, some of which will be better than others and, with a fixed number of iterations, the solution eventually obtained is essentially random within the bounds of the size of the oscillation. An effective terminating condition should be able to identify the best solutions and so increase overall performance.

The ideal termination condition is full convergence, i.e., when further iterations have no effect on the configuration of the contour. In situations where convergence does not occur rapidly enough, or at all, alternative criteria must be sought.

5.3.4 The Variational Approach

When the variational approach is used to minimise the energy of the contour, full convergence is hard to achieve. If a semi-implicit approach is used, it has been reported that, under favourable conditions, most of the elements will converge, but, cycles and chaotic behaviour can occur in part or all of the contour, [AWJ90, Sam91]. When an explicit update scheme is used, full convergence is even harder to attain. In this case, the dependence of convergence on the timestep, \( \delta t \) (see equation (62) in the previous chapter), has been noted, [Coh91, Ivi96, Sam91], and Ivins and Porrill derive a theoretical bound on the timestep for convergence in the absence of image forces:

\[
\delta t < \frac{1}{\alpha + 3\beta}
\]  

(64)

where \( \alpha \) and \( \beta \) are the coefficients of the tension and stiffness energies, respectively, see equation (61). This bound does not hold when the image forces are present however and empirical estimates of \( \delta t \sim 0.01 \) to ensure convergence in the presence of image forces have been suggested, [Ivi96]. This increases the number of iterations required to reach the region boundary considerably (by at least an order of magnitude) and is not practical in many situations. It may be possible to use an adaptive timestep whose value is only decreased once the contour nears the boundary of the region as in the work of Fua and Leclerc, [FL90], but this kind of strategy tends to rely heavily on heuristics and can be unreliable.

In order to keep the required number of iterations down to a reasonable level, it is desirable to have the timestep fixed at unity. As mentioned however, variational ARMs then become unstable in the vicinity of a true image boundary. This instability is illustrated in Figure
28. A variational ARM (CID in fact) with the timestep set to unity is applied to the task of extracting the green circle from a white background. The ARM is initialised on the blue polygon shown on the leftmost image in Figure 28 and allowed to run for 100 iterations after which time the true boundary has been reached and the ARM is in the configuration shown in red on the leftmost image. Further iterations however still affect the position of each element and the remaining diagrams show the configuration of the ARM (in red), compared to the configuration after 100 iterations (in blue), after 1, 5, 10 and 50 iterations. As well as oscillation of each individual element, which may cover a range of several pixels perpendicular to the contour boundary, a slow tangential rotation is also observed.

![Figure 28 Various configurations of a variational ARM having reached the desired image boundary.](image)

### 5.3.4.1 Termination Conditions

The points raised in the last subsection suggest that rather than seeking full convergence of ARMs based on the variational approach, finding an alternative termination condition might be more practical. An effective termination condition should allow good configurations of the model to be detected whilst keeping the value of the timestep at unity so as to allow the model to expand and contract quickly when it is far away from the region boundary.

As the ARM is driven by an energy minimisation procedure, the value of the energy is a sensible measure around which to base a terminating condition. As mentioned in section 4.4.3, Leymarie and Levine used the total contour energy as a measure on which to base a termination condition. Because of the region energy term in the energy function of the ARM, the total contour energy is difficult to calculate. However, the magnitude of the force at each element, found from (62), provides an estimate of the change in energy caused by the corresponding move made to that element. The sum of the magnitudes of these forces therefore provides an estimate of the change in energy of the contour over a particular iteration. As noted in section 4.5.1.2, however, when elements are added to or deleted from the contour, the overall energy balance of the contour is not preserved. If the sum of the force magnitudes is normalised by the number of elements in the contour to yield an average force magnitude (AFM), the measure is more consistent. An element that exhibits unusual behaviour, for example, at the edge of the image, or is close to another portion of the boundary, cannot move in the direction of the force applied to it. Such elements may have a large force applied but still be stationary and as this force may never decrease, these elements are excluded from the average.
The graph of this AFM is always roughly monotonic, see Figure 29, and so can be thresholded to yield a termination condition. Choice of a value for the threshold is difficult however. A minor problem is that once the contour has reached the region boundary, the value is far from constant and tends to vary about some mean value. The major problem however, is that both this mean value and the size of the variation about the mean value are highly dependent, although in no obvious mathematical way, on the values of the internal model parameters, $\alpha$, $\beta$, $\rho$. They are also dependent to a certain extent on properties of the region being extracted, but, as long as the properties of the region of interest and the background do not change significantly, these values are usually reasonably consistent throughout a typical region tracking task. Figure 29 shows graphs of AFM versus iteration number of a variational ARM, using a Gaussian SCM with $k$ set to 3.0, with various internal parameter settings applied to the track image of Figure 3. Because of these difficulties, a strategy for adaptively setting the threshold on the AFM while tracking an object through a sequence of images has been developed.

As a single object is tracked through a temporal sequence of images, it is assumed that conditions do not change drastically and suitable values for the threshold remain fairly constant throughout the sequence. The scheme proceeds as follows.

- A maximum number of iterations is retained in case the condition is never satisfied.

Figure 29 Typical graphs of the average force magnitude versus iteration number. $\alpha$ is fixed at 1.0 and $\beta$ and $\rho$ (the image energy coefficient) have the values given in the legend.
Chapter 5 Statistical Region Tracking

- On the first image of the sequence, the maximum number of iterations are executed and the minimum value of the AFM over all these iterations is recorded.
- That minimum value is set as the threshold for subsequent images in the sequence. Once an AFM below that value is found on a subsequent image, processing is terminated on that image and moved on to the next.
- If, on a subsequent image of the sequence, the threshold is never attained and the maximum number of iterations is executed, the threshold is increased to the minimum value of the AFM that was attained during those iterations. For safety, it is wise to put a limit on how much the threshold can be increased, for example, twice the original threshold set on the first image of the sequence. In practice, however, this safety factor is rarely necessary.

With this termination condition applied, using the number of iterations required over the tracks sequence is reduced from 5500 to 3308 and overall computation time decreases to 10 minutes and 29 seconds. The threshold found on the first image of the sequence is 5.38 and this increases to 6.15 by the end of the sequence.

5.3.5 The Energy Function

In this section some analysis of the original energy function of the statistical snake, (61), is performed in an attempt to improve the stability of the ARM. For the termination strategy described in section 5.3.4.1 to be most effective, the variation in the AFM once the ARM has reached the boundary of the region of interest should be as small as possible. This may be achieved by minimising the oscillations of each contour element. As most of the noise/jitter arises from discretisation of the image, the primary concern is with correctly calculating the force due to the image energy and no attempt is made to improve stability by examining the internal energy. As noted in section 4.4.3, however, many alternative internal energy formulations have been proposed that may improve overall performance and stability. Although no comparison of these alternative internal energy formulations is provided in this work, and Kass, et al's original formulation is used throughout, such a comparison would doubtless be useful.

5.3.5.1 Calculating the Image Force

In the OIP and CID ARMs, the image force on each element was found by calculating the goodness of a locally averaged pixel value, scaling by the coefficient, \( p \), and imposing the resultant force on a unit vector normal to the contour at that element. The locally averaged pixel value was found by averaging the pixel values along the discrete lines joining the element to its neighbours.

Consider an idealised contour, \( u(s) \), continuously parametrised by the real-valued \( s \). The image energy of \( u(s) \), when applied to an image, \( I \), is given by
Chapter 5 Statistical Region Tracking

\[ E_{\text{image}}(u(s)) = -\rho \int_{\mathbb{R}} G(I(x,y)) dx \, dy \quad (65) \]

If the contour is displaced by a small amount \( \delta u(s) \), by use of a Taylor expansion,

\[ E_{\text{image}} + \delta E_{\text{image}} = E_{\text{image}}(u(s) + \delta u(s)) = E_{\text{image}}(u(s)) + \delta u(s) \frac{dE_{\text{image}}(u(s))}{du}_{u(s)} + \text{h.o.t} \quad (66) \]

Thus to a first approximation,

\[ \delta E_{\text{image}} = \delta u(s) \frac{dE_{\text{image}}(u(s))}{du}_{u(s)} \quad (67) \]

The derivative in (67) is the rate of change of the image energy with respect to the position of the contour, which is the sum of the energy around the contour. This is proportional to the integral of the goodness around the contour. Thus,

\[ \delta E_{\text{image}} = -\rho \int_{u(s)} G(I(u(s))) ds \quad (68) \]

If the displacement is local to one part of the contour (c.f. moving a single element), the integral need be performed only over the part of the contour where the displacement is non-zero. The magnitude of the force causing such a displacement is obtained by dividing the energy change by the displacement, thus, from (68)

\[ F_{\text{image}} = \frac{dE_{\text{image}}(u(s))}{du}_{u(s)} = \rho \int_{u(s)} G(I(u(s))) ds , \quad (69) \]

where \( \delta C \) is the parameter range over the portion of the contour that is displaced.

5.3.5.2 The Discrete Averaged Goodness ARM

To find the image force on an individual discrete element, the properties of the image along the lines joining the element to its neighbours are examined as that is the portion of the boundary over which the displacement is non-zero during the move. To approximate the line integral in (69), the average along these lines is taken and used as the value at the element. This follows Ivins and Porrill's scheme apart from the fact that (69) suggests that the goodness itself should be averaged rather than the pixel values. Averaging along lines in this way, as opposed to over some neighbourhood with depth perpendicular to the contour, is appealing because it minimises uncertainty close to image boundaries. Using, for example the circular elemental neighbourhoods of Zhu and Yuille, the underlying image is blurred and localisation of the true boundary can suffer, see [ZY96].

Results can be significantly different when the local goodness measure is found in this way. The problem with the OIP method is that very different pixel values whose individual
goodnesses might be very low can be averaged to yield some pixel value close to the approximated mean of the SCM. Potentially this allows some pixels that are not part of the region of interest to be classified positively by the ARM. The variational ARM whose local goodness average is calculated by taking the average goodness instead is subsequently known as the Discrete Averaged Goodness (DAG) ARM. By inspection, the behaviour of the DAG ARM does not appear to differ greatly from that of the CID ARM and, by tuning the internal parameters, results can be obtained that are hard to distinguish. Graphs of the AFM against iteration number for the first image of the sequence are shown in Figure 30. From these graphs, it is clear that the behaviour of the two models is different and that the DAG ARM tends to attain lower AFM values once the true boundary is found (after about 220 iterations). The value of the AFM threshold selected from the first image of the track sequence for the DAG ARM is 5.06, as opposed to 5.38 for the CID ARM. By the end of the sequence, the threshold has increased to 5.59.

![Figure 30 AFM versus iteration number for CID and DAG ARMs with identical parameter settings applied to the first image of the track sequence.](image)

The DAG ARM makes multiple calls to the SCM at each contour element in contrast to the OIP and CID ARMs which make just one. The DAG ARM might therefore be expected to be more computationally expensive than the others even when a look-up table is used in the SCM (see Appendix B, section B.2). In fact the processor time required for the track sequence, with exactly the same internal parameter settings, was found to be 16 minutes and 15 seconds,
which is slightly less than the OIP and CID ARMs. This discrepancy is due to the fact that, at a fixed value of the SCM tolerance parameter, k, the CID and OIP ARMs tend to include more pixels than the DAG and hence the contour is longer and has more elements. When the termination condition of section 5.3.4.1 is applied, the overall processor time is reduced to 7 minutes and 17 seconds, which is significantly less than the OIP and CID ARMs. Because the AFM graph for the DAG ARM is smoother, termination generally occurs sooner and the total number of iterations executed is reduced to 2651, despite the fact that the AFM threshold remains closer to its original value.

Some other methods for estimating the line integral in (69) have also been attempted. More accurate estimates can be made by splitting the line into portions contained within single pixels, for example, by using a marching squares algorithm (see Appendix B, section B.5). Contributions to the whole line integral from each portion can then be found by multiplying the goodness of the pixel by the length of the portion of the line contained within it. These contributions are summed to yield the whole integral. Alternatively, as reported in [AB97a], bilinear interpolation can be used to obtain a continuous goodness function over the image, see section 5.3.6.1.3, and expressions can be derived for the line integral over portions for the line, which again are summed to yield the whole line integral. Preliminary results obtained from ARMs in which the line integrals are calculated in this way have shown that this method is not beneficial in terms of stability or overall performance; furthermore computation time is significantly increased. For these reasons, these models have been abandoned and are not considered further in this thesis, although they are discussed to some extent in [AB97a] and, for completeness, the algorithms for obtaining these line integrals are included in Appendix B section B.5.

5.3.6 Alternative Energy Minimisation Schemes

In section 4.4.4, alternative energy minimisation schemes that have become increasingly popular for use with ACMs were discussed. In this section, the application of the two most popular alternative energy minimisation schemes, greedy algorithms and dynamic programming, both of which can guarantee full convergence of an ACM, to ARMs is investigated. Other energy minimisation techniques discussed in that section have not been applied to these models. Although their application is equally plausible, they offer no clear advantage in terms of convergence, computational overhead, or ease of implementation over the techniques that are considered in this thesis.

5.3.6.1 Greedy Algorithms

The application of greedy algorithms to the minimisation of the energy of active contours was discussed in section 4.4.4.2. Each element is updated in turn and to compute the move of a
particular element, the overall energy of the contour is computed with that element placed in each of a set of candidate new positions. The position that reduces the overall energy furthest is selected as the new position for that element.

In contrast to the potential energy in ACMs, the region energy in an ARM cannot be divided into contributions from each element and so estimation of the overall energy change that occurs when a single element is moved is less straightforward. The internal energy change is still simple to calculate using the finite difference expressions, given in equations (54) and (55), for the first and second derivatives in the tension and stiffness energy terms, respectively. If the $s$-th element moves from $u'(s)$ to $u''(s)$, by summing the energy changes at elements $s-1$, $s$, and $s+1$, the changes in tension and stiffness energies, respectively, are found to be

$$
\delta E_{\text{ten}} = 2\alpha \left[ \frac{1}{2} \left( u''(s) - u'(s) \right) - \frac{1}{2} \left( u'(s) - u'(s-1) + u'(s+1) \right) \right]
$$

and

$$
\delta E_{\text{sti}} = 2\beta \left[ \frac{3}{2} \left( u''(s) - u'(s) \right) - \left( u'(s) - u'(s-2) - 2u'(s-1) - 4u'(s+1) + u'(s+2) \right) \right].
$$

Figure 31 shows three consecutive elements of a contour, $u$, on an image are shown as is the set of candidate new positions for the middle element. The triangles that would be included (or excluded) by one of these moves are shown.

However, since the image energy is the integral of the goodness within the contour, see equation (61) in section 4.5.1, the change in the image energy that is caused by a single element move is the integral of the goodness over the two triangular regions depicted in Figure 31. Figure 31 shows three consecutive elements of a contour, whose positions, although stored discretely in greedy implementations, are depicted at the top left-hand corner of the pixel at which they are located. The central element of the three is to be moved to one of the positions in the 3x3 square with its current position at the centre. The set of candidate new positions is outlined in the figure and each pixel represented by a candidate position is numbered. An example of a potential move is shown where the contour element moves upwards by one pixel, from candidate position 4, which corresponds to no move being made, to candidate position 3. Two triangles are created by this move and the goodness contained within those triangles represents the change in image energy of the contour that is caused by such a move. There are
Chapter 5 Statistical Region Tracking

various ways to obtain an approximation for the integral of the goodness over such regions and three approaches are used in this thesis, which are presented in the rest of this section.

5.3.6.1.1 A Simple Greedy ARM

To obtain an estimate of the image energy change, a similar approach to that used in the variational ARMs can be used. The simplest and computationally cheapest approach is to calculate the average goodness along the lines to each neighbour from the initial position of the element. Then, for each candidate position, calculate the area, signed to indicate expansion/contraction of the boundary, of the two triangles created by the old position, the new position and each of the neighbouring elements. The average of the two goodness values, weighted by the corresponding areas and scaled by the coefficient, $p$, then gives an approximation to the change in image energy caused by that move:

$$\delta E_{\text{image}} = \rho \left( \frac{\int_{s-1}^{s} G(I(u(s))) \, ds}{L(u(s-1), u(s))} + \frac{\int_{s}^{s+1} G(I(u(s))) \, ds}{L(u(s), u(s + 1))} \right)$$

(71)

$\Delta_{s-1}$ and $\Delta_{s+1}$ denote the areas of the two triangles (see Figure 31) and $L$ denotes the length of the line, which here is the number of pixels found by the Bresenham line drawing algorithm.

Greedy ARMs implemented in this way, known henceforward as Single Line Greedy (SLG) ARMs, are considerably more stable than variational ARMs. Although the models do not generally fully converge, many of the contour elements will reach stable positions and most others enter simple oscillations and cycles. Complex cycles and chaotic behaviour can still occur, however, particularly when the image energy coefficient, $p$, is large compared to $\alpha$ and $\beta$. Figure 32 shows a typical steady state of a SLG ARM once the true boundary has been located; configurations of the contour after five consecutive iterations are shown. Simple oscillations and cycles can be observed in some portions of the contour, while at others convergence is achieved.

Sometimes the steady state cycles of these models can be simple enough to detect and, as well as full convergence, a further termination condition is enforced. If the same number, which must be less than ten, of elements moved in each of the last ten iterations, then a simple cycle is declared and processing is terminated. In both cases, the number ten is chosen as a value for which false detection is very rare, while most simple cycles are detected in reasonable time. The steady state shown in Figure 32 constitutes a detectable cycle as four contour elements move at each iteration. There are therefore four different types of steady state for these models: full convergence, detectable cycles, undetectable cycles and chaotic behaviour. These

120
states can be visualised from graphs of the number of elements that are still moving against iteration number and Figure 33 shows graphs for each of these different steady states.

![Diagram showing typical steady state of a SLG ARM](image)

**Figure 32** Typical steady state of a SLG ARM showing convergence of some elements and some simple oscillations and cycles.

![Graphs showing number of moving elements against iteration number](image)

**Figure 33** Graphs of number of moving elements against iteration number showing the four different steady states.

When applied to the track sequence with the usual parameter settings, the SLG ARM never fully converges and simple cycles are detected on only 6 of the 26 images; a total of 4717 iterations are performed. This model is faster than any of the variational models, despite the increased number of iterations, and requires only 6 minutes 0 seconds of processor time – approximately 13 iterations per second.
5.3.6.1.2 An Improved Strategy

With the single-line approach above, the image energy approximation is inconsistent in the sense that the image energy change approximation that is calculated for a move from one position to another is different to that calculated when moving back. Consider the opposite move to the one being made in Figure 31, which is depicted in Figure 34. The central element's position is one pixel higher than in Figure 31 and so to move back to the original configuration it must move down one pixel to candidate position 5. Although the triangular regions are the same, the goodness is now averaged along the lines joining the higher position to the two neighbouring elements and so the value of each average is not the same as that calculated for the opposite move. The overall energy change approximation is thus not equal to minus the approximation calculated for the opposite move, as would be expected.

Some consistency can be added by calculating separate goodness averages for each candidate move. Single element oscillations can be avoided if the image energy change approximation can be constructed in such a way that the change in energy caused by a particular move is minus the change caused by moving back to the original position. By carefully selecting the end points of the lines along which the goodness is averaged for each candidate position, this condition can be satisfied. Consider the 3x3 square pixel neighbourhood with the current element position at the centre. If these candidate positions are enumerated as shown in Figure 31 and Figure 34, then, when calculating the goodness average for each candidate move, the end points for the lines along which the averages are taken that are suggested in Table 4 can be used. There are other possible choices, but the one given suffices. These averages are again weighted by the areas and scaled by $\rho$ to yield the energy change, as in (71).

To see how this works, consider the example given in Figure 31 and Figure 34. For the upward move in Figure 31, the candidate new position under consideration is position 3. From Table 4 then, the goodness average for the left-hand triangle is calculated by averaging the goodness along the line from the previous element to the new position (candidate 3). For the right-hand triangle, however, the goodness is averaged along the line from the current position...
(candidate 4) to the next contour element. When the opposite move, shown in Figure 34, is considered, again from Table 4, for the left hand triangle the goodness is averaged along the line from the previous element to the current position, which is the same line used in the original move. Similarly, the goodness for the right hand triangle is averaged over the line from the new position and so is the same as for the opposite move. When the areas of the triangles are calculated, they will be minus the areas calculated for the move in the opposite direction and so the overall energy is minus that for the opposite move. It is thus impossible for the element to move back to the original position. The energy change caused by the upward move must have been negative, otherwise another candidate position, in particular candidate 4 (no move), would have been preferable. The energy change caused by the return move must therefore be positive and so another candidate position must be selected.

ARMs using this approximation to the image energy change are referred to in the remainder of this thesis as Multi-Line Greedy (MLG) ARMs. The multi-line scheme is computationally more expensive, as a goodness average must be computed for each candidate move, but the convergence properties of the model are enhanced. Although two-step single element oscillations of the type shown in Figure 32 cannot now occur, it is still possible for single element cycles of more than two steps, or cycles involving more than one element to occur. When an MLG ARM is applied to the track sequence, full convergence is achieved on 3 of the 26 images and on a further 7, detectable cycles occur. 3927 iterations were required, in total, which took 8 minutes and 36 seconds of processor time – approximately 8 iterations per second.

<table>
<thead>
<tr>
<th>Candidate</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>End point of line to previous element</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>-</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>End point of line to next element</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>-</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4 Lists the end points of the lines joining to the previous and next elements of the contour, along which the goodness is averaged to ensure single step consistency of the image energy change approximation.

5.3.6.1.3 Fully consistent approximations to the image energy change

For full convergence to be guaranteed, the energy change calculated when the contour deforms from one configuration, through any number of other configurations and back to the original, must be zero, i.e., any sequence of configurations, \( u_i \), \( i = 1, \ldots, n \), such that \( u_1 = u_0 \), must satisfy the following condition:

\[
\sum_{i=1}^{n-1} \delta E(u_{i+1}, u_i) = 0. \tag{72}
\]

Using a marching squares process around the boundary of the triangular regions created by a single element move, refer to Figure 31, each triangle can be decomposed into a set of
polygonal regions each completely contained within one pixel, which correspond to the intersection of the triangle with that pixel. The goodness contained within the triangle can then be evaluated by summing the area of the intersection of the triangle with each pixel multiplied by the goodness of that pixel. When the energy change is approximated in this way, condition (72) is satisfied and so all oscillations and cycles of the type discussed above are completely eliminated. It is still possible for very complex cycles involving repeated addition and deletion of elements to occur, but such occurrences have proved extremely rare. Although these cycles can be contrived and have been observed on images used in this thesis for extreme settings of the energy term coefficients (very large $\rho$), at the moderate settings most commonly used in practice they rarely occur.

The image energy change calculated in this way is exactly the surface integral of the discrete goodness function, defined over the discrete image, over the triangular regions created by single element moves. However, it is still referred to as an approximation to the true image energy change. It is considered an approximation because the image itself is a sampled approximation to an underlying continuous image function. A closer approximation to this underlying function can be obtained by interpolating the discrete values given in the image, for example, by bilinearly interpolating the image. In a similar way a closer approximation to the true, continuous goodness function over the image can be found by interpolating the discrete function obtained by calculating the goodness of each individual pixel. By using such an approximation to the true goodness function, even better approximations to the true image energy change can be obtained. Here, a bilinearly interpolated goodness function is used. Expressions for the surface integral of this function over upright rectangular and right-angled triangular sub-pixel regions are not hard to derive and, by decomposing the extended region over which the surface integral is required into such atomic components, the surface integral over the whole region can be found. Expressions for these integrals are provided, mainly for the reader wishing to emulate this technique, with derivations, together with the marching squares algorithm used to decompose triangular image regions into atomic components, in Appendix B, section B.5.

The computational load for ARMs using this technique, henceforward referred to as Surface Integral Greedy (SIG) ARMs, is high and such a model may only be useful as a yardstick against which to measure performance of other simpler models. However, it is worth noting that the energy change calculations for each candidate new position, for both the SIG ARM and the MLG ARM of the last subsection, may be carried out in parallel to a granularity determined by the size of the elemental neighbourhood. Over the usual track sequence, full convergence is attained on every image and the total number of iterations is reduced considerably to only 1702, but the processor time required is 88 minutes and 23 seconds – an
average of 3 seconds for each iteration. Although this ARM and the other greedy ARMs are clearly more stable than the variational ones, again parameters can be tuned in such a way that, although some difference can be seen between the results obtained, it is hard to select which results are the best. Figure 35 shows the final configurations of DAG and MLG ARMs, applied to the images of Figure 3, with exactly the same internal parameter settings as were used in the OIP ARM in Figure 26. Neither of these pairs of contours appears to be as accurate as the OIP ARM contours in that figure. The reason for these poor results is that appropriate settings for the internal parameters are different for different models. These results illustrate the inadequacy, for comparing performance, of results that simply consist of the contours obtained by applying different implementations to the same image with the same parameter settings. If the tolerance parameter, k, is increased to 3.5, more similar results are obtained. Contours obtained using the MLG ARM with k set to 3.5 are shown in Figure 36. These contours, particularly that for the track, are visibly different to the OIP contours but it is hard to say which are the best. The most distinctive differences occur in the bottom right hand corner of the track region. The OIP ARM misses part of the region at the bottom of the image, which is found by the MLG ARM, however, the OIP ARM moves further into the slightly darker region on the far right at the bottom than the MLG.

Figure 35 Contours extracted by DAG (top) and MLG (bottom) ARMs from the images of Figure 3.
Chapter 5 Statistical Region Tracking

Figure 36 Contours extracted by the MLG ARM with k increased to 3.5.

5.3.6.2 Dynamic Programming

The image energy change estimation techniques used to implement greedy ARMs could also be used to implement dynamically programmed ARMs as the same basic problems apply. Convergence again would only be guaranteed if energy changes are calculated consistently, so that condition (72) is satisfied, i.e., the goodness surface integrals discussed in section 5.3.6.1.3 must be employed. With limited resources such a model is not feasible.

Another drawback of the dynamic programming approach, although it does allow inclusion of hard constraints into the model, is that the positions of the elements must be updated together after all the moves have been calculated. This complicates the avoidance of self-intersection. For these reasons, dynamically programmed formulations of ARMs have not been implemented.

5.4 Summary

In this chapter, three different classes of region tracking algorithm have been investigated:

• Thresholded likelihood images – no strict topology is enforced on the positively classified image region, but if that region is reasonably compact, results can be improved by the use of a morphological post-processing step. Computation times are manageable and do not depend on parameter settings, but computation is significantly increased by incorporation of the post-processing step. Convergence is trivial.

• Region growing – a loose assumption of connectedness of the region interest is made. When the region is compact, again, results can be improved by a morphological post-processing step. The computational load is reasonable for moderate parameter settings but can become very heavy for extreme values of the underlying statistical model tolerance parameter, k. The boundary smoothing achieved by incorporation of the post-processing step helps to keep computation times down. Again convergence is trivially assured.

• Active region models – the positive region is strictly connected and, by virtue of the internal constraints on the smoothness of the contour, is reasonably compact. For simple
implementations, the computational load is relatively small, but convergence of the algorithm is not guaranteed. To ensure convergence of the algorithm, the computational load must be greatly increased. Table 3 at the beginning of this chapter provides a summary of the different ARMs that have been implemented and tested in this work.

Variants of all three types of algorithm are very common in segmentation and tracking systems but very little comparison of their performance has been reported. Intuitively, the ARM seems the most suitable algorithm for classifying pixels as in or out of a region of interest when such regions are known to be connected and compact, as they are here. Furthermore, stability and convergence intuitively seem like desirable properties for the ARM to possess. However, without quantitative comparison, definitive statements concerning which algorithms, or variants thereof, are best for a particular task cannot be made.

In the next chapter, methods for quantitatively measuring the performance of these algorithms are investigated. Recall from Chapter 3 that, as well as comparing the different region tracking algorithms discussed in this chapter, measures of the performance of these algorithms are also required to provide a comparison of the underlying statistical colour models. The algorithms are thus viewed as classification procedures and their performance is assessed accordingly.
Chapter 6 Performance Characterisation

In the first chapter of this thesis, three questions were posed that need to be addressed while investigating how best to exploit colour data for the applications of interest in this thesis. The first question - how best to characterise an object in terms of the colour data that corresponds to it, was addressed in Chapter 2 and Chapter 3. In Chapter 3 a number of different statistical models for colour data were presented. The second question - how best to exploit these models in order to perform region tracking tasks, was addressed in Chapter 4 and Chapter 5. In Chapter 5 a number of region tracking algorithms based on underlying statistical models were presented. Attention is now turned to the third question - how can these models and region tracking algorithms be compared so that the best combination can be identified?

In this chapter a general discussion of performance assessment and characterisation is given. Existing techniques for the evaluation of segmentation and region segmentation algorithms are then briefly reviewed. As discussed in previous chapters, a region tracking algorithm may be regarded as a device for obtaining a classification from a statistical colour model. Accordingly, a measure of performance of region segmentation or tracking systems based on receiver operating characteristic (ROC) analysis (see Appendix C) is proposed. Finally, an experimental protocol based on this measure for obtaining a comparison of the performance of different statistical models and algorithms is outlined.

6.1 Performance Characterisation in Computer Vision

For many years, the issue of performance characterisation of machine vision algorithms was neglected. As noted by Haralick [Har94], the primary concern of machine vision researchers when the field was relatively new was to demonstrate that vision tasks could be performed automatically at all. Exactly how well that task was performed and that one method performed better than another was of minor importance. Results tended to consist of raw algorithm output from just one or two images to be interpreted by inspection. Often, the reader was left to compare output of the algorithm with an intuitive ideal.

As early as 1991, Jain and Binford, [JB91], lamented the lack of experimental tradition in the field of machine vision. More rigorous experimentation and the development of evaluation techniques for specific problems and applications were called for in order that the field may begin to grow as a scientific discipline. Jain and Binford also suggested a rationalisation of the field by defining common vision tasks, e.g., segmentation, edge detection, etc, specifically in terms of their inputs and outputs. Such a rationalisation should promote the development of algorithm independent evaluation techniques and hence greatly simplify the job of a vision engineer trying to build full systems from such components. Chellappa and
Rosenfeld, [CR93, Ros93], adopted these ideas and generated a set of rules of vision engineering emphasising the importance of carefully identifying both the problem being addressed and the domain of application.

In [Har94], Haralick goes on to describe an experimental protocol for characterising the performance of machine vision algorithms. He proposes the use of a parametric model of the range of input data to which the algorithm is designed to apply. A test set for a specific application is constructed by varying each of these parameters over their range and sampling the variation in image content and added noise. Another set of parameters exists within the algorithm itself and Haralick suggests that a performance characterisation is obtained by averaging the performance over the test set at each combination of the values of any other parameters defined within the algorithm itself. The measure of performance at each combination of internal parameter settings, or operating points, is obtained using a self-assessment criterion within the algorithm that decides whether or not the algorithm output on a particular input is reliable. Failure and false alarm rates of this error criterion are found at each operating point and these results characterise the overall performance of the system.

Thacker, et al, [TC92, CTC96], suggest that rather than modelling the world in which the algorithm is to work, as in Haralick's protocol, the algorithm itself should be modelled in terms of its modes of failure. Expressions for the probabilities of different failures can be generated in terms of algorithm parameters and probabilities of different input types. These may be used to find an expression for the overall probability of failure of an algorithm that can be evaluated for a particular application by estimating the probabilities of different types of input data, and minimised with respect to the internal algorithm parameters. In replies to the discussion on his paper, [Har94], and in subsequent work, [Har96], Haralick also uses algorithm models to allow noise models to be propagated through algorithms to different levels of larger systems. Such models then allow the estimation of the performance of an entire system from a low level noise model, which is useful when designing large systems and trying to choose the best combination of components for that system.

6.2 Existing Segmentation Assessment Schemes

Many different measures have been proposed to provide measures of quality of the output of segmentation algorithms. As region segmentation and tracking are special cases of the general segmentation problem, see section 4.1, at least in terms of input and output, many of these schemes are also applicable to these classes of algorithm. Zhang, [Zha96], provides a review of existing segmentation assessment techniques in which he identifies three different classes of scheme:

- **Analytic methods** examine the theoretical aspects of an algorithm and attempt to derive a measure of performance of the algorithm in terms of the principles upon which it is based.
Chapter 6 Performance Characterisation

- **Empirical goodness methods** provide measures of quality of algorithm output by examining properties of the output that are intuitively favourable. Such measures are most commonly used within algorithms, for example, as discussed in section 2.3.1, Ohlander, et al, and Ohta, et al, [OPR78, OKS80], used a measure of the difference between data in the region of interest and background data to automatically select threshold values. Many other measures, such as measures made on the shape of the region of interest, have been used for similar purposes and a review of automatic thresholding techniques is provided by Sahoo, et al, [SSW88]. Another use for such measures of quality is as a termination condition for ACMs, an example is the termination condition proposed by Berger, [Ber91], discussed in section 4.4.3.

- **Empirical discrepancy methods** derive measures of output quality by comparison of the algorithm output with an external ideal output or "ground truth". The ground truth data is generated by some external means, for example, for real-world images by outlining each region of interest by hand or for synthetic data directly from the system that generates the synthetic data.

6.2.1 Pros and Cons

Analytic methods provide a very fast comparison and can be useful for eliminating certain choices without even having to implement an algorithm. However, little information about the quality of the algorithm output can be ascertained and so, in general, they do not provide sufficient information upon which to base a final decision.

Empirical goodness methods clearly provide a better indication of the quality of the algorithm output than analytic methods as in the latter the output is not examined at all. The necessity for implementation of each algorithm to be assessed means that the performance measures are not obtained as quickly as they are from analytic methods. However, results are more quickly obtained than they are from discrepancy measures as the need for ground truth data is avoided. As objective measures of performance, however, these methods are not ideal. In much the same way as for the information criteria, such as the AIC, discussed in section 3.8, in order for such a measurement to be relied upon, the results of its use must be compared to results obtained from an evaluation scheme that is less vulnerable to bias, such as an empirical discrepancy measure. Recently, Palmer et al, [PDK96], proposed a performance measure for boundary detection algorithms and, in order to justify its use, showed that results obtained tallied well with performance measures derived from comparison with ground truth data. However, the role of this type of segmentation assessment measure remains predominantly as a measure for making on-line decisions rather than to generate precise measures of overall performance of an algorithm.

By virtue of the use of ground truth data for comparison, which cannot be incorporated into an algorithm, empirical discrepancy methods provide a much higher degree of objectivity
in their performance assessments. The obvious drawback of these methods is that ground truth data must be produced for every member of the set of images over which the performance is to be characterized. This can be a very expensive process, particularly if a test set of real images is used, as the ground truth must be accurate in order not to introduce bias into the evaluation. In the work of Solloway, et al, [STH96], for example, each ground truth image has to be generated, by hand, by an expert radiologist.

6.2.2 Other Methods

There are some other ways to measure the performance of segmentation algorithms that have been used in the past but that do not fit into any of the three categories described above. An important measure is computation time. Unlike measures of computational complexity, which may be classed as an analytic measure, in order to measure actual computation time, implementation of the algorithm is necessary and so this measure cannot be classed as purely analytic. However, no examination of the algorithm output is necessary in order to generate the measure, which therefore does not fit into either of the other two categories. Nonetheless, computation time is often a crucial consideration in the choice of algorithm, affecting not only the resources required by the algorithm in the eventual work place but also the time required for training and testing.

Leung and Lam, [LL96], performed a performance analysis of a number of iterative thresholding techniques over two sets of data, one real and one synthetic. Measures of performance – the probability of convergence, the average number of iterations required for convergence and a measure of the consistency of these measures, were generated over each set. A further measure for each set was also used. For the synthetic data set, a probability of error, described in the next section, was generated by comparison with a ground truth image. For the real data set, a subjective evaluation is also included which provides a count of the percentage of algorithm outputs that are deemed satisfactory by a human observer. This latter “measure” adheres to the dubious assertion of Pal and Pal, [PP93], that human observation is the best way to evaluate and compare segmentation algorithm output.

Another measure is used by Graaf, et al, [GKV92], to evaluate segmentation algorithms that are not expected to produce perfect output. It is assumed that some pre or post processing will have to be done by hand in order to bring the automatic segmentations up to a usable standard. The amount of time taken over this processing is used as a performance measure of the original segmentation algorithm.

Finally, to compare the performance of a number of edge detection algorithms, Bryant and Bouldin, [BB79], suggested a performance comparison which could also be used for segmentation algorithms, [Zha96]. Rather than using a hand generated ground truth image, the consensus results of all the other algorithms under question is used. For algorithms with near
perfect performance, this may produce a reasonable comparison, but for less perfect algorithms or more difficult problem domains, the results are less reliable. In either case it is difficult to know if and when the comparison can be trusted.

6.3 Empirical Discrepancy Methods

In this work, a measure of (region) segmentation quality is required so that informed decisions can be made about which algorithms and models are best, or most appropriate under particular conditions. To ensure that any conclusions made are not biased in any way, there is little choice but to use an empirical discrepancy measure. Many methods have been developed for comparison of automatically generated image segmentations with ground truth segmentations and once more Zhang, [Zha96], divides the available techniques into three categories:

- **Pixel based methods** measure the discrepancy between an automatically generated segmentation and ground truth by examining pixels that are misclassified according to the ground truth. The most direct way to generate a measure of performance in this way is simply to count the number of misclassified pixels and generate a probability of misclassification. Lee, et al, [LCP90], generate a probability of error, $p(E)$, in exactly this way to evaluate binary region segmentations.

$$p(E) = p(O)p(B|O) + p(B)p(O|B).$$  \hspace{1cm} (73)

Where $p(B|O)$ is the probability of an object pixel being classified as background and $p(O|B)$ is the probability of a background pixel being classified as object, both of which are obtained directly by comparison of output with ground truth. $P(O)$ and $p(B)$ are the a priori probabilities of being part of the object and the background respectively. This definition is not hard to extend to multi-class problems, see [DH73], such as full segmentation. Campbell, et al, [CMT97], see section 2.3.2, use this kind of measure to assess the performance of their region classification system, but also break the misclassification counts down into a confusion matrix. Other similar methods, see [Zha96] and references therein, incorporate the positions of misclassified pixels to generate more complex measures, which take into account the severity of each misclassification.

- **Over and under segmentation** are important factors when assessing full segmentations and correspondence between the number and position of regions in the automatic and ground truth segmentations have been used frequently to generate overall measures of quality of a segmentation. Some such measures are described and referenced by Zhang, [Zha96], and, more recently, Campbell, et al, [CMT95], and Hoover, et al, [HJJ96] have proposed more sophisticated measures of this type. However, when the number of regions is pre-specified as in region segmentation, over and under segmentation is of no concern and so is not considered further in this thesis.
Properties of regions can be compared between the automatic segmentation and ground truth and a measure of similarity of pairs of corresponding regions can also be used to provide a measure of quality of a segmentation. If the true number of regions is unknown, as in the general segmentation problem, it is difficult to apply this kind of approach because of the problems arising from over segmentation and under segmentation although they can be combined as in [HJJ96]. For problems in which the number of classes is known, such as region segmentation and tracking, however, pairs of regions are readily identified and so this kind of method is more suitable. Typically, measures of size and shape are made from each region and compared to yield a measure of similarity or dissimilarity of the regions. A simple measure of the difference in shape of two regions is given by the Hausdorff distance between the two sets of pixels that comprise them. Huttenlocher, et al, [HKR93], proposed the use of the Hausdorff distance, which can be efficiently computed by use of a distance transform, [Bor86], for such purposes. A related approach, used by Gunn and Nixon, [GN97], is the use Fourier descriptors, [PF77], c.f., the ACMs of Staib and Duncan described in section 4.4.5.2, to describe the shape of the two regions to be compared. The Euclidean distance between the vectors of basis function weights is used as a measure of discrepancy. If further information is known about the positions of actual landmark points, comparison of the positions of these points can be made to generate discrepancy measures. Solloway et al, [STH96], measure the distance of corresponding points on the boundaries of the automatically extracted region and the ground truth segmentation. The mean and variance of the distance are used to obtain a discrepancy measure.

Pixel misclassification is an attractive option because of the definition of region tracking algorithms that has been used in previous chapters, in which they are regarded as pixel classification procedures. The task of a statistical colour model is to distinguish colour pixels that are part of the region being modelled from those that are not. Assessment of such a model directly, in terms of how often that task is correctly performed and how often it is not, therefore seems appropriate and is the class of performance assessment scheme that will be adopted. Incorporation of positional information into misclassification measure does not offer any obvious improvement in this context.

Methods based on comparisons of region properties are less directly related to the problem in question. A secondary issue, which also applies to pixel misclassification methods that incorporate positional information, is that computation time for each comparison is increased by having to perform analytical calculations for each extracted region. Direct comparison of two regions and counting misclassifications, on the other hand, is cheap. This point is particularly salient for tracking applications when the number of region comparisons required is large.
6.4 Other Considerations

Before discussion of how best to generate actual measures of performance from counts of misclassifications, there are some other general concerns to be addressed and decisions to be made with regard to application of such a performance assessment scheme. These considerations are the subject of this section.

6.4.1 Ground Truth

One of the main aims of this work is to model naturally arising distributions of colour data and so the data sets used for comparison of the performance of the models must be comprised of real rather than synthetic data. However, this introduces the laborious process of obtaining ground truth data, which must be generated by hand.

Pal and Pal, [PP93], observed in their review of fuzzy segmentation techniques that boundaries in real images are rarely definite. Usually, a boundary region exists around the edge of a region in an image and contains pixels whose classification is unclear. If a single ground truth region is defined by hand with its border within this boundary region, each pixel within the boundary region is classified as in or out of the region of interest. This can introduce bias to a discrepancy measure, as other regions whose borders also lie completely within the boundary region, can be unfairly penalised with misclassifications. To avoid this, the ground truth needs to account for the uncertainty around borders of image regions. Ground truths are constructed here by defining an inner boundary and an outer boundary. Both boundaries classify pixels as “in” or “out” of the region of interest. Those classified as “in” by the inner boundary are pixels about which there is no doubt that they belong to the region of interest. Those classified as “out” by the outer boundary are pixels about which there is no doubt they do not belong to the region. The classification of the remaining pixels is said to be unknown. A misclassification then occurs only if a pixel inside the inner boundary is classified as “out” or a pixel outside in the outer boundary is classified as “in”. Ground truth images thus consist of three distinct regions, as shown in Figure 37, which shows ground truth data for the two images shown in Figure 3.

Ground truth data produced in this way are not only more robust than binary ground truth data but may also be cheaper to obtain. Neither boundary requires the degree of accuracy to which a single binary ground truth segmentation must be produced. In particular, very small loss or penalty costs may be assigned to pixels in the doubt region being classified by the system as inside or outside the region. If the loss or penalty costs are zero, pixels classified in this way do not contribute to the performance measure at all so very little is lost by excluding some “in” pixels from the inner segmentation and including some “out” pixels from the outer segmentation. This is the approach adopted here and is similar to the approach used by Ferri and Vidal, [FV92]. The characterisation obtained may be more optimistic than one derived from a
ground truth in which every pixel contributes. However, this is consistently the case so the comparisons of performance made are still fair, in fact much fairer than comparisons of performance measures derived from binary ground truth images would be.

![Figure 37 Ground truth data for the two images shown in Figure 3.](image)

### 6.4.2 Limits of Accuracy

When applying performance characterisation techniques of this type, the limits of accuracy of the ground truth must always be borne in mind. For example, if the ground truth data is generated by hand as discussed in section 6.4.1, the fact that the ground truth is not perfect must be acknowledged. A limit on the accuracy of performance measures generated by comparison with ground truth data generated by hand can be obtained by generating a performance measure of an alternative set of hand generated data. As long as performance measures of algorithms are significantly lower than the performance of the second set of ideal data, the performance measures may be considered valid. However, when algorithm performance is comparable to that performance, the accuracy of the ground truth must be called into question and results cannot be relied upon.

Even when the measure of performance of a system falls well within this limit there may be factors that can introduce uncertainty into the measure. Possible sources of such errors should be identified so that an estimate of the uncertainty in measures of performance can be made. Such an estimate can then be used to identify when the difference in measured performance of two systems is significant and when it is not.

### 6.5 Performance of Classification Systems

There are a number of ways in which a performance measure can be generated by consideration of the classification of individual pixels. These options are considered in this section.

#### 6.5.1 Probability of Error and Bayes Risk

The simplest performance measure that can be derived from a straight comparison of an automatic segmentation with a ground truth image is the probability of error. Using Bayes rule
for the simple two-class case of region segmentation or tracking, this is given by equation (73).
For more general, multi-class, segmentation problems, this definition is not hard to extend, see [DH73].

Simply using the probability of error as a performance measure assumes that the importance of different types of misclassification is equal. In many applications this may not be the case. Suppose a particular algorithm can be used to detect a certain class of objects in a scene. Typically, such an algorithm will formulate some measure of likelihood that such an object exists at each point in an input image. This likelihood might then be thresholded to yield a binary decision about the existence of an object at each point in a similar way to the likelihood images of section 5.1. As the threshold level is decreased, the number of missed objects decreases but the number of false alarms increases. Suppose the algorithm is used as part of a system where its function is simply to find one of the class of objects that it detects. As it is unimportant which particular object is selected the detriment in performance incurred by a missed object is minimal but the potential detriment incurred by a false alarm is large. Conversely, if one particular object of that class is sought by the system, some other module will have to examine the properties of each object detected by the algorithm and select the most likely. Missed objects may seriously affect overall performance, but false alarms are less important and may only effect the overall computation time. In the first scenario, the threshold should be set high but in the second it should be set low.

Simply using the probability of error does not provide sufficient information in general. Once a particular application area has been identified, a loss can be assigned to each of the two types of misclassification (missed object and false alarm). Using the values of these losses, the Bayes risk can be defined, [DH73], by weighting the probabilities of each type of error by its associated loss value. The Bayes risk provides a more general measure of performance than the probability of error.

6.5.2 Receiver Operating Characteristic (ROC) Analysis

By varying an internal parameter that controls the probability of occurrence of each type of misclassification, missed object (false negative) and false alarm (false positive), and plotting values depending on the counts of these occurrences against each other, a number of points on an ROC curve can be obtained. Often, as in this work, the ratio of true positives to the total number of positives is plotted against the ratio of false positives to the total number of negatives. The curve then lies in the unit square and must pass through the origin, where no cases are positively classified, and the point (1, 1), where every case is positively classified, see Figure 38 (right). In an ideal system, for some value of the internal parameter it is possible to classify all cases correctly so that the true positive ratio is one and the false positive ratio is zero. The curve then also passes through the point (0, 1).
Chapter 6 Performance Characterisation

A simple model of a classification algorithm is often used, in which it is assumed that the algorithm makes some measure on each case (or pixel) to be classified and the decision is made by thresholding on the value of that measurement. If the values of that measurement for each type of case (positive and negative) are drawn from separate probability distributions, then the classification power of the algorithm depends on how well separated these two probability distributions are. This model is illustrated in Figure 38. In the picture on the left, the distributions of values of a measurement obtained on the positive and negative cases are depicted. That picture also shows how application of a decision threshold, whose position is determined by the value of an internal algorithm parameter, gives rise to different true and false positive ratios. As the position of the decision threshold is varied, these ratios also vary and, if plotted against each other, give rise to ROC curves such as those shown in the figure on the right. In a typical system, the two distributions will overlap, as shown in the figure on the left, giving rise to an ROC curve like that shown in blue in the figure on the right. If the two distributions are completely separated, perfect performance is obtainable by placing the decision threshold between the two distributions and the resulting ROC curve passes through the point (0, 1), like the curve shown in pink in the figure on the right. If the two distributions are coincident, the decisions made by the algorithm are equivalent to random chance and the ROC curve lies along the “chance diagonal”, as shown in green in the figure on the right. If the distributions remain fixed as the value of the internal parameter controlling the position of the decision threshold is varied, the ROC curves must be monotonic, as they are in Figure 38.

For simple algorithms like the thresholded likelihood image algorithm of section 5.1, this model is perfectly accurate, as the measure made on each pixel is the likelihood, which is thresholded directly in the algorithm. For more complex algorithms, such as an ARM, however, the measure made by the algorithm on each pixel is not easily expressible. As well as the likelihood assigned to each pixel, factors such as position in the image, likelihood of neighbouring pixels and the settings of the internal ARM parameters, also contribute.

![Figure 38 Illustration of the common algorithm model used in ROC analysis](image-url)

Some useful measures of performance can be derived from an ROC curve. Intuitively, the area under the curve should increase as the overall classification power of the system
increases moving the curve closer to that of an ideal system. The area, $A$, under the curve provides a measure of the degree of discrimination between the positive and negative cases found in the ground truth. In particular, if the values of the measurements made on each type of pixel are drawn from normal distributions, then the area can be shown, [Swe79], to be a monotonic function, $z^1$, of the distance between the means of the two distributions scaled by their variances. Thus,

$$z(A) = \frac{(\mu_1 - \mu_2)}{(\sigma_1^2 + \sigma_2^2)^{1/2}},$$

where $\mu_i$ and $\sigma_i$ denote the means and variances, respectively, of the two distributions and $z$ is the inverse standard cumulative normal distribution function. $z$ is thus a monotonic function of $A$, so that even if the normal assumption is not justified, order is preserved in a performance comparison.

This measure is obtained without assigning a loss to any type of misclassification and thus is parameter free. When an algorithm is to be employed by a particular application, values and losses are assigned to the different decision outcomes. An appropriate setting for the decision threshold parameter is then available directly from the ROC curve by finding a point on the curve with the appropriate slope, see Appendix C, where a more detailed description of ROC analysis is given.

**6.5.3 Doubt-Misclassification Curves**

ROC analysis is designed for the characterisation of binary decision making procedures. In more general cases, where there are more than two possible classifications, although it may be possible to compute an ROC curve for each pair of classes, other methods of analysis may be more appropriate. Ripley, [Rip96], suggests that the number of cases assigned to a doubt category might be plotted against the total number of misclassifications. This assumes the existence of a doubt category in the output of the algorithm and, in a similar way to ROC analysis, the existence of some parameter that controls the strictness of decisions. As the strictness of decisions is increased, more cases will be assigned to the doubt category. As it is decreased, the algorithm is forced to make decisions for pixels about which it is less sure and so the misclassification rate increases.

Such a characterisation suffers from the same problem as the simple probability of error discussed in section 6.5.3, in that no loss is assigned to different types of misclassification. It is not clear how performance measures independent of Bayes loss could be derived from such a characterisation.
6.6 A Measure of Performance

As the region tracking algorithms used here can be regarded as binary decision making procedures, ROC analysis can be used to generate measures of performance for them.

In order to generate a performance measure for an algorithm applied to a particular set of data an ROC curve for that algorithm must be generated over that set of data. According to the definition of section 4.1, a region tracking algorithm produces a region segmentation for each image to which it is applied. For all the algorithms used here, these region segmentations are binary. Each of these region segmentations can be compared to the corresponding ground truth image to obtain true positive and false positive pixel counts. These counts are summed over every image of a data set to yield one point on the ROC curve. By varying the value of some internal parameter of the algorithm, for example, the statistical model tolerance parameter, k, more points on the ROC curve can be generated. The set of values over which the parameter is varied defines a set of operating points at which performance is assessed. These values must be selected in order to ensure a good coverage of the ROC curve.

Once such a set of points on the ROC curve has been computed, they can be interpolated to approximate the true curve. An estimate of the area under the curve can then be made and a corresponding estimate of the measure of performance, $z(A)$ of equation (74), obtained. In this work, the area is estimated by using a trapezium rule, which provides a slightly conservative estimate of the overall performance, since the ROC curves are usually convex.

6.7 Experimental Protocol

The preceding section describes how a measure of performance can be generated for one particular algorithm, over a particular set of data and with fixed settings for all internal parameters other than the one used to parametrise the ROC curve. In order to generate a meaningful performance measure for an algorithm that is suitable for comparison with similar measures obtained from other algorithms, the other internal parameters must be set to suitable values. In this section, an experimental protocol based on the performance measure described in the last section is outlined. For a particular application, this protocol can be used to select appropriate values for the internal parameters of algorithms; select the most effective statistical model; and, further, provide a comparison of the performance of a set of different algorithms.

6.7.1 Internal Parameter Optimisation

The most direct way to find an optimal set of internal parameter settings is to produce ROC curves over the set of input data for each possible combination of parameter values. In practice, only a finite set of combinations can be tested and so a suitable range, over which each internal parameter setting is varied, must be determined by preliminary experimentation. The
Chapter 6 Performance Characterisation

A combination of parameter settings that produce the ROC curve that indicates the highest performance is then selected.

In many cases, there may be a large number of such internal parameters that need tuning so the number of experiments required for full optimisation is prohibitively large. If that is the case, either some subset of the parameters must be selected over which the optimisation is to be performed, or a less complete optimisation, as suggested by Haralick, [Har94], can be performed rather than such an exhaustive search. Haralick's optimisation is performed by selecting a set of initial values for the parameters and then maximising the performance as each one is varied in turn until all the parameters have been set. In the experiments performed in the next chapter, the former approach is adopted and ranges for a subset of the available parameters are defined over which every combination is tested.

6.7.2 Training, testing and validation

Typically, when designing a system for a real world application, there will be a certain amount of data available for use in the development phase. As outlined in Haralick's experimental protocol, to be useful, this data should be a representative sample of the domain of data the system is likely to meet in the eventual workplace. To avoid bias in performance measures generated over this set, the set must be partitioned into a training set and a test set, both of which, again, should ideally be representative of the entire data domain. The training set is used to optimise the internal parameters of the algorithm, as discussed in the previous subsection. The final performance characterisation is then performed over the previously unseen test set. The ROC curve generated over the test set is also used to find settings for the decision parameter once values and costs of the decision outcomes have been specified.

Ripley, [Rip96], suggests that a further partition can be made so that there is, in addition to the training and test sets, a validation set. This set is used in tandem with the training set to check that the settings found during training generalise well to other data of the same type, i.e., still produce good (if not optimal) performance. If they do not, something is wrong, either in the range of the training set or in the algorithm itself and further investigation is necessary. However, due to the lack of data for which ground truths have been generated, no validation is used in the experiments detailed in the next chapter.

6.7.3 Summary

The final part of this section provides a summary of the experimental protocol used here to compare the performance of different algorithms using different underlying statistical models.

1. Obtain training, test, and, if used, validation, sets representative of the domain of input data likely to be encountered in the eventual application.
Chapter 6 Performance Characterisation

2. Construct ground truth data, as discussed in section 6.4.1, for all training, validation and test data.

3. Select an internal parameter of the algorithm to parametrise the ROC curves.

4. Perform some preliminary experimentation to find value ranges for the remaining internal algorithm parameters to be optimised.

5. Choose the set of operating points, i.e., the set of values to be used for the parameter selected in 3. Methods for selecting these values are discussed in section 7.2.3.

6. For each combination of internal parameter settings, run the algorithm over the training set at each operating point and count the numbers of false positives and true positives at each. Thus, obtain an ROC curve for each combination of internal parameter settings.

7. Estimate the area under each ROC curve generated in 6 and calculate the measure of performance, $z(A)$, see equation (74) in section 6.5.2.

8. Select the internal parameter settings that produced the highest performance measure.

9. If a validation set is used, fix the values for the operating points and the ranges of the other internal parameters and perform a similar set of experiments over the validation set. If the optimal parameter settings found over the training set are also optimal over the validation set, or at least have comparable performance to the optimal settings, accept those settings as optimal. If the optimal settings from the training set perform significantly worse than the best settings over the validation set, check the theory behind the algorithm, that the algorithm is implemented correctly and that the training and validation sets are large enough to be representative of the application domain. Return to step 1.

10. Run the algorithm over the test set of input data to produce one more ROC curve. The estimate of $z(A)$ obtained from this ROC curve is the overall measure of performance of the algorithm over the given set of data.

11. To select the best model or algorithm, choose the one that produced the highest performance measure at stage 10. An operating point for the algorithm for a particular application is found by assigning costs and values for different decision outcomes, see Appendix C.

No part of the overall methodology is specific to the algorithms used here. Very similar methods could be used to characterise the performance of most decision type algorithms provided there is a parameter available to parametrise the ROC curves. Also, apart from the ground truth extraction, there is no part of the methodology that could not, in theory, be fully automated in a performance characterisation module for region tracking/segmentation algorithms or other classes of algorithm.

6.8 Discussion

Although the general principles are similar, the use of ROC analysis for performance evaluation and parameter setting differs from using Bayes risk in one fundamental respect. If ROC analysis
is used, values and losses to the various types of decision outcome are not assigned until the very last stage when the algorithm is to be applied in some system. When Bayes risk is used, these values and losses are set from the start. Once an ROC curve has been produced for a particular algorithm with a particular set of internal parameter settings (other than the one used to parametrise the ROC curve itself), over a particular set of data, no further performance analysis is required. If at some later stage, the values and losses are revised, the performance measure still stands and the new value for the parameter controlling the ROC curve is immediately available from the existing curve. Using Bayes risk, strictly, the entire performance analysis including setting of other internal parameters over a training set and then finding measures of performance over the test set for each candidate algorithm, see 6.7, must be performed again. Often reliable estimates for the values and losses may not be available in the early stages of algorithm selection and so the use of ROC is more conservative and likely to be more robust.

From this fundamental difference, another important difference arises. If ROC analysis is used, the measure of performance is generated before values and losses are assigned. Other internal parameters are optimised by choosing the values that produce the highest measure of performance. Once values and losses are assigned and a suitable value for the ROC curve parameter has been found, an estimate for the expected Bayes loss can be found from the ROC curve. This estimate will not necessarily be the same as if the Bayes loss were used as the performance measure from the start. When the Bayes loss is directly minimised over the training set, the optimal parameter settings may not be the same as those given by the area under the ROC curve, which maximise the overall degree of discrimination. If the training set provides a good representation of the entire application image domain, so that the level of performance over the test set (and any other unseen data) is similar to that obtained over the training set then the optimal parameter settings suggested by the Bayes risk analysis are likely to perform better in the workplace than those generated by the ROC analysis. However, in most cases, the training set is less comprehensive and performance on unseen data is worse than the optimised performance over the training set. This suggests that in fact the optimal values found over the training set are not the optimal values over the entire image domain, but, assuming the training set is of reasonable size, they should be close. In this kind of situation, the values found using ROC analysis are likely to be more reliable as the measure incorporates information about performance at similar parameter values, at least of the ROC curve parameter, which may be more suitable for some of the data in the image domain.

The experimental protocol used here is similar to that proposed by Haralick, but is based on a measure taken directly from the classification performance of the algorithm rather than characterising the performance in terms of self-assessment. The use of models of both the
environment and the algorithm has been avoided throughout and direct empirical assessment of real world performance on real data is preferred.
Chapter 7 Experiments and Results

In the preceding chapters, a number of statistical models have been proposed for naturally occurring distributions of colour data and a number of algorithms that exploit these models for region tracking have also been proposed. In Chapter 6 a performance measure for region tracking algorithms was selected and an experimental protocol, based around that measure, was outlined which can be used to compare and select region tracking algorithms and underlying statistical models within a particular application. This chapter presents details of the application of that procedure to two data sets, which represent two separate region tracking applications. The chapter begins with a description of the two data sets over which the experiments are performed and then some general experimental considerations and preliminary results are discussed. A series of experiments in which the three types of region tracking algorithm discussed in Chapter 5 - thresholding, region growing and ARMs, are compared is then presented. During this comparison, the performance of the various ARM implementations proposed in Chapter 5 is assessed, thus providing a comparison of the performance of these different models. This comparison is used to select one ARM to be used in an experimental comparison of the statistical colour models proposed in Chapter 3. A comparison of the statistical colour models is performed for each application (data set) using the selected ARM, together with the simple thresholding algorithm for verification.

7.1 Data sets

Two separate data sets, representing two separate applications, are used in this work. Apart from one sequence, all the images used were obtained from a hand held Kodak DC50 digital camera with the image quality set high, the flash turned off and all other settings at default values. These images are 24-bit, 756x504 colour images. The other sequence, provided by DERA Chertsey, was obtained using a Panasonic E550 camera, an MIL standard video recorder and a Parsytec CFG frame-grabber. These images are 24-bit, 512x512 colour images. Ground truth images for every image of each sequence from each data set have been generated, by hand, as described in section 6.4.1. The hand outlines are defined by using a mouse to mark vertices in the image, which are joined automatically to form a polygonal approximation to the boundary.

7.1.1 Clothes set

The first set of images consists of sequences of images containing highly non-rigid, single coloured objects of interest - items of clothing, in cluttered surroundings. There are three sequences of images, each containing roughly twenty images, two of which contain two regions of interest, and so effectively count as two separate sequences, and one that contains only one.
Chapter 7 Experiments and Results

A single illuminant is used in an otherwise darkened room and very little surface reflection occurs from these objects. The objects of interest are moved a small distance between consecutive frames of the sequences and the illumination and viewing directions are fixed. These conditions adhere strongly to the assumptions inherent in the unichromatic reflection model on which the directional chromaticity models are based and the purpose of this data set is to determine which of these models is best.

![Figure 39](image.png)

**Figure 39** The first image of each sequence from the clothes data set. The seed regions are superimposed on each region of interest.

7.1.2 Tracks Set

The second set of data consists of sequences of images containing roads and lanes of varying surface quality, all obtained under daylight illumination. For all but one of the sequences, the bottom right in Figure 40, the vantage point for each image is head height above the ground and the viewing direction is approximately horizontal looking parallel to the direction of the road at the vantage point. However, there is some variation in pitch, yaw and roll as each image was taken by hand. The distance between vantage points for consecutive images of the sequences is approximately 2 metres. For the other sequence, the vantage point is slightly lower as the camera was mounted on the roof of a car. There is less variation in viewing direction with respect to the ground plane in this sequence, since each image was not taken by hand but was extracted from the sequence of images obtained directly from the camera. The distance covered between consecutive frames is also approximately 2 metres. None of the roads are occluded at any time in any of the sequences.
There are six sequences in total ranging in length from 15 images to 67, and containing a total of around 200 images, see Figure 40. This data set is designed to mimic the data that might be obtained from a camera mounted on the front of an autonomous cross-country vehicle.

![Figure 40 The first image of each sequence from the tracks data set. The seed region is superimposed on each.](image)

### 7.2 General Considerations

This section discusses some general considerations that apply to all the experiments presented in this chapter.

#### 7.2.1 Seed Regions

Each region tracking algorithm requires a predefined seed region in the first image of each sequence from which to construct a SCM. For the region growers and ARMs, the seed region is further required to specify the initial configuration. For the tracks data set, there is a certain
similarity in the position, shape and size of the region of interest throughout. The region of interest is always roughly triangular, with its base at the bottom of the image and roughly centred horizontally in the image. This similarity can be exploited to find a suitable seed region that can be used consistently and does not have to be defined by hand at the beginning of each sequence. This generic seed region was constructed by examining the intersection of all the inner ground truth regions over each 756x504 image in the tracks data set. A polygonal seed region was drawn by hand within the region of the image array that is within at least 90% of the inner ground truth regions. This seed region is shown in each image of Figure 40. For the smaller image, the polygon is translated horizontally so that its position relative to the centre of the image is the same.

The positions of the regions of interest in the clothes data set possess no such consistency and a separate seed region is required for each. The seed regions were defined by hand for each region of interest, but once defined were stored so that the same seed is always used with the same region of interest. The seeds are of roughly the same size for each region of interest and are defined by rectangular contours, near the centre of the region, in the first image of the sequence.

7.2.2 Training and Test sets

From each of the two data sets, two sequences were taken as training sets, over which to find optimal internal parameter settings, and the rest retained as the test sets. For the clothes data set, the first (top left) image sequence of Figure 39, which contains two regions of interest, comprised the training set. For the tracks data set, the first (top) two sequences of Figure 40 comprised the training set. No validation set was used in either case.

7.2.3 Operating Point Selection

A good choice of parameter for the ROC curves is the SCM tolerance parameter, $k$, defined in equation (17), as it directly affects the numbers of true and false positives. As $k$ is increased, the goodness of each individual pixel, and thus its chances of being classified as part of the region of interest, is also increased. Values for this parameter to define operating points must be determined before experimentation can begin. Ivins and Porrill, [IP95, Ivi96], suggest that, for a Gaussian statistical model, values of $k$ can be read directly from a chi-squared table if the required percentage of true positives is known. Even for the Gaussian model, however, this has proved unreliable in practice. Moreover, no such methods are available for some of the other SCMs, particularly the Bingham based ones. This idea has therefore been abandoned in favour of empirical selection of the values of $k$ for operating points.

The heuristic used here, to select $n$ operating points, is the following: choose moderate settings for other internal algorithm parameters. Find an upper setting for $k$ such that the false
positive ratio is approximately \((n-1)/n\). Find a lower setting such that the true positive ratio is approximately \(1/n\). Find a moderate setting such that the false positive count is approximately equal to false negative count. Interpolate between these three points to find values for the other operating points. This ensures good coverage of the ROC curve.

In all the experiments described here, 15 operating points are used. Values of \(k\) for these operating points are found by applying the above heuristic over the data in the training set of a particular data set. The values must therefore be set separately for each set of data. Rather than interpolating linearly between the three values found by this initial experimentation, as the most interesting part of the curve tends to be around the central operating point, a quadratic interpolation is used that causes the operating points to bunch around the centre. For example, using a Gaussian SCM, with a DAG (discrete averaged goodness, see section 5.3.5.2) ARM over the tracks training set, the three values of \(k\) found using the method above were 1.35, 3.0 and 10.5. The intermediate values are calculated so that the step sizes between them increase linearly as they get further from the central point, as shown in Table 5.

<table>
<thead>
<tr>
<th>Operating Point</th>
<th>Value of (k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.35</td>
</tr>
<tr>
<td>2</td>
<td>1.76</td>
</tr>
<tr>
<td>3</td>
<td>2.11</td>
</tr>
<tr>
<td>4</td>
<td>2.41</td>
</tr>
<tr>
<td>5</td>
<td>2.65</td>
</tr>
<tr>
<td>6</td>
<td>2.82</td>
</tr>
<tr>
<td>7</td>
<td>2.94</td>
</tr>
<tr>
<td>8</td>
<td>3.0</td>
</tr>
<tr>
<td>9</td>
<td>3.27</td>
</tr>
<tr>
<td>10</td>
<td>3.8</td>
</tr>
<tr>
<td>11</td>
<td>4.61</td>
</tr>
<tr>
<td>12</td>
<td>5.68</td>
</tr>
<tr>
<td>13</td>
<td>7.01</td>
</tr>
<tr>
<td>14</td>
<td>8.63</td>
</tr>
<tr>
<td>15</td>
<td>10.5</td>
</tr>
</tbody>
</table>

Table 5 Calculation of operating points.

For the likelihood thresholding algorithm, varying the threshold has a similar effect to varying the SCM tolerance parameter. To decrease overall computation times, the threshold is used to parametrise the ROC curves and \(k\) is fixed for each SCM at the central operating points used for the ARMs. The threshold is varied over the range \(\{15, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180, 195, 210, 225\}\), again giving 15 operating points.
Chapter 7 Experiments and Results

7.2.4 Significance Levels

Some noise is introduced into the estimate of \( z(A) \) by the discrete nature of the ROC curve from which the area under the curve is estimated by use of a trapezium rule. For the same ROC curve there will be some variation in the measures that would be obtained if slightly different operating points were chosen. To measure the size of this variation, an ROC curve over the tracks training set was generated using a Gaussian SCM and DAG ARM at three times the experimental resolution, i.e., with 45 operating points. The additional operating point values were slight perturbations, one above and one below, of each of the values in Table 5. A random sample of 10000, 15 point ROC curves, where one operating point from each set of 3 is randomly selected were generated and the area under that curve was measured. The standard deviation of the area over that sample of curves was found to be 0.00001. Henceforward, all areas, \( A \), calculated in this way will be assumed to lie within the range \( A \pm 0.00001 \). The corresponding error in \( z(A) \) is found by transforming the extremes of the range of areas by the same function, see section 6.5.2.

7.2.5 Limits of significance

One limit on the significance of the performance measure used here can be derived from the error margins found in the previous section. As the area under an ROC curve becomes close to one, because \( z(A) \) tends to infinity as \( A \) tends to one, the error margin of \( \pm 0.00001 \) becomes increasingly significant. Thus, one clear limit beyond which the measure cannot be relied upon is \( z(0.99999) = 4.265 \). Another limit, however, can be obtained by analysing the integrity of the ground truth, as discussed in section 6.4.2.

Over a small subset of the tracks training set (10 consecutive images of one sequence), two volunteers were asked to extract the region of interest from each image by hand with the task specified in five different ways. The first time, every effort must be made to exclude all negative cases and so only pixels about which there is no doubt that they are part of the track are included in the region. The second time, a conservative estimate to each contour is required, in which the approximation should be fairly accurate but taking care not to include any pixels outside the region. The third time as close an approximation to the true contour is requested. The fourth time a liberal estimate is asked for, which again is quite accurate but care is taken not to exclude any pixels that are part of the region. For the final task the contour must be drawn in such a way as to ensure that all positive pixels are included. The resulting sets of binary images are compared to the original hand defined ground truth images to obtain 5-point ROC curves in the usual way. The resultant ROC curves are shown in Figure 41. As performance is extremely high, only the top left-hand corner of the graph is shown so that the discrepancy from perfect performance can be appreciated.
Figure 4.1 ROC curves from hand region segmentations.

The areas under these curves were found to be 0.999993 and 0.999952, which give values of 4.3 and 3.9, respectively, for z(A). At the central operating point, the misclassification rate is about 0.05% in both cases. Taking the smaller of these two values, it is subsequently assumed that performance measures of 3.9 or over cannot be used to decide that one system produces better performance than another in this application. However, the results below show that this level of performance is never attained by any of the region tracking algorithms using any statistical model.

7.3 Comparison of Region Tracking Algorithms

In this section the results of a series of experiments to compare the performance of the three types of region tracking algorithm discussed in Chapter 5 are presented. The comparison is carried out over the tracks data set only. A preliminary comparison between SCMs, [AB97], has shown that the trivariate Gaussian SCM performs well over this data set and so it is used throughout this comparison. The comparison follows the experimental protocol outlined in section 6.7. Thus a training phase occurs first in which the internal parameters of each algorithm are varied and the combination that produces the best performance is selected as optimal. A final performance measure is then obtained over the test set using the optimal parameter settings.

7.3.1 Thresholded Likelihood Images

Other than the SCM tolerance parameter, k, and the threshold, the only tuneable parameter of this algorithm is the size and shape of the structuring element used in the morphological post-
processing step. This step consists of a morphological closure and a morphological opening, but it is not clear in which order these two steps should occur. Only circular structuring elements are investigated and the same structuring element is used for each of the two morphological steps.

Performance of the algorithm is measured over the training set as the diameter of the structuring element is varied over the range \{1, 3, 5, 7, 9, 11\}, where 1 corresponds to having no post-processing step. The performance measures are plotted in Figure 42. In the figure, "Cl, Op" indicates that the morphological operation consisted of a closure followed by an opening, whilst "Op, Cl" indicates opening followed by closure.

![Figure 42 Plot of performance of the thresholded likelihood image algorithm over the tracks training set as the diameter of the structuring element is varied.](image)

In Figure 42, the highest performance measure is obtained for the post-processing step consisting of closure followed by opening with a structuring element of diameter 9 pixels. When the operations are applied in the reverse order, the optimal structuring element diameter is found to be 7 pixels. With the optimal post-processing step, the algorithm is applied to the test set and a final ROC curve is computed for which \(z(A)\) is found to be 1.739. The minimum misclassification rate over the set of operating points is about 9%. The ROC curve is shown in Figure 48, in section 7.3.4, with ROC curves from the other two types of algorithm.
7.3.2 Region Growing Algorithm

The SCM tolerance parameter is used to parametrise the ROC curves generated using the region growing algorithm. The same morphological operation used with the thresholded likelihood images is applied to the raw algorithm output from each image and the diameter of the circular structuring element is varied over the same range. Performance measures obtained over the tracks training set, using a Gaussian SCM, are plotted in Figure 43.

For the region growing algorithm, the highest performance is obtained when the post-processing step consists of closure followed by opening with a structuring element of diameter 11 pixels. From the chart in Figure 43, it appears that the optimal structuring element may have diameter greater than 11. Because of the heavy computation required to generate ROC curves for the region growing algorithm (roughly 48 hours continuous processing each), however, no investigation beyond the original range has been performed and so 11 is selected as the optimal structuring element diameter. The algorithm with the selected post-processing step is applied to the tracks test set to compute a final ROC curve, which is shown in Figure 48. From this ROC curve, $z(A)$ was found to be 1.896 and the minimum misclassification rate was about 6%.
Chapter 7 Experiments and Results

7.3.3 ARMs

This section contains an experimental comparison of the different ARM implementations discussed in section 5.3. The ARMs have many internal parameters that can be tuned and it is not practical to perform an exhaustive search for the best combination of settings for them all. The only parameters that will be optimised experimentally, using the protocol described in section 6.7, are the coefficients of the three different terms in the ARM energy function, (61), $\alpha$ - the tension energy coefficient, $\beta$ - the stiffness energy coefficient and $\rho$ - the image energy coefficient.

For the variational ARMs, the timestep, $\delta t$, of equation (62), is fixed at unity, which causes the displacement calculated for each element to almost always be greater than one. The displacement thus gets normalised, as discussed in section 4.5.1, and so there is a linear dependence between the values of the three coefficients. For the greedy ARMs, the displacement is always 1 pixel and so the linear dependence always exists. The addition and deletion of elements to and from the contour reduces the effect of the tension energy term the purpose of which is to minimise stretching of the boundary, i.e., the distance between consecutive elements of the contour. For these reasons, in practice, the tension energy coefficient is fixed, to a value of 1.0, and only the two remaining coefficients are optimised.

For all the other internal parameters, reasonable values are found, as described in this paragraph, and are fixed for all the experiments reported in the remainder of this thesis. To set the thresholds on the distance between consecutive elements of the contour before new elements are added and deleted, the hand defined ground truth polygons were examined. In order to give the ARM the capability to sample those polygons accurately, these thresholds are set in such a way as to ensure that approximately twice the number of contour elements exist in the contour as vertices in the ground truth polygon when the area within the contour is roughly the same as within the ground truth polygon. The minimum distance threshold was set to 3 pixels, as whenever two neighbouring elements are within this distance, the self-intersection avoidance scheme ensures that one of them is deleted anyway. A value of 15 pixels for the maximum threshold was found to produce approximately the right result. A maximum number of iterations of 500 was found to ensure that no further significant movement of the contour will occur on the first image of each sequence. On subsequent images of each sequence, this value is reduced to 200 as the difference between initial and final configurations is smaller.

Some preliminary experiments must be performed to find a suitable range over which to vary the values of the two parameters to be optimised. A wide range of values for each parameter is selected - $\beta \in \{0, 0.05, 0.1, 0.3, 0.5, 0.7, 0.9, 1.0, 3.0, 5.0\}$ $\rho \in \{0.9, 1, 3, 5, 7, 10, 20, 30\}$. With each combination of values, the DAG ARM was applied to the tracks training set to obtain an ROC curve from which the performance measure, $z(A)$, was computed. Values of
Chapter 7 Experiments and Results

$z(A)$ for each combination are plotted in Figure 44. The maximum performance is obtained when $\beta=0.5$ and $p=5.0$ and these values appear to be in the region of where the optimal settings lie. The graphs also show that performance deteriorates sharply when $p$ is one or less and noticeably when $\beta$ is set to zero.

![Figure 44 Plot of $z(A)$ for the DAG ARM applied to the tracks test set with a wide range of parameter settings.](image)

With $\beta=0.5$ and $p=5.0$, the set of operating points for each ARM are calculated (and recalculated for the DAG ARM) in the way described in section 7.2.3. The values of the SCM tolerance parameter, $k$, used for the low, middle and high operating points, for each ARM are given in Table 6.

<table>
<thead>
<tr>
<th>Op pt.</th>
<th>OIP and CID ARMs</th>
<th>DAG ARMs</th>
<th>Greedy ARMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>0.8</td>
<td>1.35</td>
<td>1.4</td>
</tr>
<tr>
<td>Mid</td>
<td>2.6</td>
<td>3.0</td>
<td>3.2</td>
</tr>
<tr>
<td>High</td>
<td>10.0</td>
<td>10.5</td>
<td>10.5</td>
</tr>
</tbody>
</table>

Table 6 Operating points (values of $k$) used for each ARM using a Gaussian SCM over the tracks data set.
Chapter 7 Experiments and Results

**OIP ARM, no termination condition**

- $p = 3$: 3.46772, 3.46677, 3.46725, 3.46317, 3.44992, 3.44031
- $p = 5$: 3.44918, 3.44549, 3.46472, 3.44658, 3.46755, 3.46563
- $p = 7$: 3.39175, 3.40061, 3.43224, 3.46013, 3.4558, 3.45814

**OIP ARM, with termination condition**

- $p = 3$: 3.46944, 3.47099, 3.11121, 2.98747, 3.16777, 3.2996
- $p = 5$: 3.46242, 3.47132, 3.48027, 3.45755, 3.44987, 3.44156
- $p = 7$: 3.45269, 3.444745, 3.41102, 3.45776, 3.47472, 3.44706

**CID ARM, no termination condition**

- $p = 3$: 3.68726, 3.66569, 3.66823, 3.46235, 3.45061, 3.44856
- $p = 5$: 3.46917, 3.45661, 3.46424, 3.46333, 3.46275, 3.46252
- $p = 7$: 3.39208, 3.40491, 3.45234, 3.46013, 3.45563, 3.45933

**CID ARM, with termination condition**

- $p = 3$: 3.46981, 3.48221, 3.47172, 3.44213, 3.42115, 3.42526
- $p = 5$: 3.46254, 3.47416, 3.47021, 3.43987, 3.42277, 3.43726
- $p = 7$: 3.44892, 3.44517, 3.46789, 3.46172, 3.47205, 3.46503

**DAG ARM, no termination condition**

- $p = 3$: 3.46502, 3.48486, 3.46829, 3.45893, 3.45432, 3.45448
- $p = 5$: 3.46839, 3.47261, 3.47374, 3.47957, 3.48009, 3.47104
- $p = 7$: 3.44489, 3.43794, 3.46195, 3.46414, 3.40644, 3.47331
- $p = 9$: 3.43007, 3.42739, 3.44654, 3.46214, 3.45515, 3.43262

**DAG ARM, with termination condition**

- $p = 3$: 3.48471, 3.49002, 3.45896, 3.46623, 3.39114, 3.34053
- $p = 5$: 3.46761, 3.48509, 3.49188, 3.43807, 3.45651, 3.47539
- $p = 7$: 3.48797, 3.48227, 3.47936, 3.45872, 3.48856, 3.46653
Chapter 7 Experiments and Results

Figure 45 Performance measures obtained from each ARM over the tracks training set.

<table>
<thead>
<tr>
<th>ARM</th>
<th>$\beta$</th>
<th>$\rho$</th>
<th>Max $z(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAG</td>
<td>0.1</td>
<td>3.0</td>
<td>3.49502</td>
</tr>
<tr>
<td>MLG</td>
<td>0.5</td>
<td>7.0</td>
<td>3.49426</td>
</tr>
<tr>
<td>DAG (NT)</td>
<td>0.1</td>
<td>3.0</td>
<td>3.48486</td>
</tr>
<tr>
<td>CID</td>
<td>0.1</td>
<td>3.0</td>
<td>3.48221</td>
</tr>
<tr>
<td>OIP</td>
<td>0.1</td>
<td>3.0</td>
<td>3.4799</td>
</tr>
<tr>
<td>CID (NT)</td>
<td>0.3</td>
<td>3.0</td>
<td>3.46882</td>
</tr>
<tr>
<td>OIP (NT)</td>
<td>0.05</td>
<td>3.0</td>
<td>3.46772</td>
</tr>
<tr>
<td>SLG</td>
<td>0.7</td>
<td>9.0</td>
<td>3.43906</td>
</tr>
</tbody>
</table>

Table 7 Optimal parameter settings for each ARM found on the tracks training set.

<table>
<thead>
<tr>
<th>ARM</th>
<th>$\beta$</th>
<th>$\rho$</th>
<th>$z(A)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAG</td>
<td>0.1</td>
<td>3.0</td>
<td>2.26209</td>
</tr>
<tr>
<td>SLG</td>
<td>0.7</td>
<td>9.0</td>
<td>2.25643</td>
</tr>
<tr>
<td>MLG</td>
<td>0.5</td>
<td>7.0</td>
<td>2.23299</td>
</tr>
<tr>
<td>DAG (NT)</td>
<td>0.1</td>
<td>3.0</td>
<td>2.20442</td>
</tr>
<tr>
<td>SIG</td>
<td>0.5</td>
<td>7.0</td>
<td>2.19982</td>
</tr>
<tr>
<td>CID</td>
<td>0.1</td>
<td>3.0</td>
<td>2.16565</td>
</tr>
<tr>
<td>OIP</td>
<td>0.1</td>
<td>3.0</td>
<td>2.16529</td>
</tr>
<tr>
<td>CID (NT)</td>
<td>0.3</td>
<td>3.0</td>
<td>2.11096</td>
</tr>
<tr>
<td>OIP (NT)</td>
<td>0.05</td>
<td>3.0</td>
<td>2.10169</td>
</tr>
</tbody>
</table>

Table 8 Performance measures obtained for each ARM over the tracks test set.
Chapter 7 Experiments and Results

Each ARM is then applied to the training set with each combination of parameter settings from the ranges $\beta \in \{0.05, 0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\rho \in \{3, 5, 7, 9\}$, in which the optimal parameter settings are assumed to lie. The performance measures obtained over the tracks training set are shown in Figure 45. In that figure, the series corresponding to the value of $\rho$ for which the highest performance was produced is marked with error bars, which show the uncertainty of the measure (see section 7.2.4). Table 7 shows the settings for each ARM that produced the highest performance over the training set, in that table and subsequent figures and tables, (NT) denotes the fact that the termination condition, of 5.3.4.1, was not used on a variational ARM. The optimal parameter settings are subsequently fixed and each ARM is applied to the test set. Table 8 shows the performance measures that were obtained over the tracks test set and these values are plotted in Figure 46, which shows that the highest performance was obtained from the variational DAG ARM with the termination condition applied. Figure 47 shows the ROC curves that were obtained from each ARM over the test set. The ROC curves in Figure 47 are very similar and so only the top tenth of each ROC curve is shown so that differences can be discerned, the ROC curves for the CID and OIP ARMs over the test set are almost identical and so only one is shown. The minimum misclassification rates attained by these algorithms range from 2.5%, for the DAG ARM with the termination condition, up to around 4%, for the OIP ARM with no termination condition.
Chapter 7 Experiments and Results

No training phase was performed for the SIG ARM because of the high computational load it imparts. The optimal settings of the MLG ARM were used to generate the ROC curve over the test set, shown in Figure 47, and the corresponding performance measure.

Figure 47 ROC curves obtained for each ARM over the tracks test set.

### 7.3.4 Comparison

ROC curves generated over the tracks test set, using a Gaussian SCM, for the optimal algorithm of each type are shown in Figure 48.
Chapter 7 Experiments and Results

Figure 48 ROC curves from the best algorithm of each type generated over the tracks test set using a Gaussian SCM.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Z((A))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete Averaged Goodness, ARM, see section 5.3.5.2, with the termination condition, of section 5.3.4.1, applied.</td>
<td>2.262</td>
</tr>
<tr>
<td>Region growing algorithm, of section 5.2, with a morphological post-processing step consisting of a closure followed by an opening with a circular structuring element of diameter 11 pixels.</td>
<td>1.896</td>
</tr>
<tr>
<td>Thresholded likelihood image technique, see section 5.1, with a morphological post-processing step consisting of a closure followed by an opening with a circular structuring element of diameter 9 pixels.</td>
<td>1.739</td>
</tr>
</tbody>
</table>

Table 9 Performance measures for each region tracking algorithm generated over the tracks test set using a Gaussian SCM.

The algorithms are summarised and their performances over the tracks test set are given in Table 9. The table shows that the ARM algorithm produces the highest performance in terms of pixel classification; the region growing algorithm is second best and the thresholded likelihood algorithm produces the worst performance of the three.
7.4 Comparison of SCMs

This section describes the experiments and presents the results comprising a comparison of the performance of the different SCMs over the two data sets. The main comparison is performed using an active region model, since, in the last section, this type of algorithm was found to exploit an underlying SCM most effectively for pixel classification. However, a further comparison is performed using the thresholded likelihood technique, without any morphological post-processing step, in order to investigate the consistency of the comparison between different algorithms.

7.4.1 ARMs

In the comparison of the ARM implementations, described in section 7.3.1, the DAG ARM with the termination condition applied was found to produce the highest test set performance. It is therefore selected for use in the comparison of the underlying SCMs. Some discussion and justification for this choice is provided in the next chapter.

The comparison proceeds, in the same way as the comparison of the ARM implementations in the last section, by following the experimental protocol described in 6.7. Thus, a training phase occurs first in which the optimal ARM parameter settings are found over the training set for each SCM. Once these parameters have been selected, they are fixed and a final performance measure is made over the test set. The number of experiments required to produce performance measures for every combination of chromaticity and intensity models is prohibitively large and so the search for the optimal model for each data set has been somewhat constrained. Firstly, using the same model for the chromaticity, the three intensity models, presented in section 3.4, are compared. Once the best intensity model has been identified, it is used exclusively to obtain a comparison of the different directional models.

The internal algorithm parameters are the same as in the ARM implementation comparison, but the ranges over which the stiffness and image energy coefficients are varied during training are restricted to $\beta \in \{0.05, 0.1, 0.3, 0.5\}$ and $\rho \in \{3, 5, 7\}$.

7.4.1.1 Clothes Data Set

The set of experiments over the clothes data set is divided into two stages. Firstly, each intensity model is tested in conjunction with a Bingham chromaticity model. Next, using the intensity model that produced the best results, the directional chromaticity models are compared with each other and with the trivariate Gaussian SCM.

7.4.1.1.1 Intensity Model Comparison

The ARM is trained using a Bingham chromaticity model with uniform, Gaussian and planar surface intensity models to find the optimal parameter settings. Plots of the performance
measure for each parameter setting combination are not very informative but are shown in Appendix D, section D.1, for completeness. Table 10 shows the maximum training phase performances, the corresponding ARM parameter settings, and the test set performance measures for each intensity model. ROC curves generated over the test set, using the optimal parameter settings found in the training phase, are shown in Figure 49. The uniform intensity model is found to produce the highest performance.

Figure 49 ROC curves generated by the ARM over the clothes test set using various intensity models

<table>
<thead>
<tr>
<th>Intensity Model</th>
<th>SCM Abbreviation.</th>
<th>( \beta )</th>
<th>( \rho )</th>
<th>Max ( z(A) ) – training set.</th>
<th>Final ( z(A) ) – test set.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>BCUI</td>
<td>0.05</td>
<td>5.0</td>
<td>2.25508</td>
<td>1.7615</td>
</tr>
<tr>
<td>Planar surface</td>
<td>BCPSI</td>
<td>0.5</td>
<td>5.0</td>
<td>2.24965</td>
<td>1.05204</td>
</tr>
<tr>
<td>Gaussian</td>
<td>BCGI</td>
<td>0.1</td>
<td>3.0</td>
<td>2.22582</td>
<td>1.05125</td>
</tr>
</tbody>
</table>

Table 10 ARM optimal parameter settings and performance measures obtained for each intensity model over the clothes training and test sets

7.4.1.1.2 Chromaticity Model Comparison
A uniform model of the intensity component is adopted throughout the following comparison of the directional chromaticity models. As usual, optimal ARM parameters are selected for each
Chapter 7 Experiments and Results

model during a training phase and final performance measures are generated using those settings over the test set. Plots of the performance of the ARM over the clothes training set using each chromaticity model, and also the trivariate Gaussian model, can be found in Appendix D, section D.2. Table 11 shows the maximum training phase performance, corresponding optimal parameter settings and test set performances for each model.

![Figure 50 ROC curves obtained from the ARM over the clothes test set using various chromaticity models.](image)

<table>
<thead>
<tr>
<th>Chromaticity Model</th>
<th>SCM Abbreviation</th>
<th>$\rho$</th>
<th>$\beta$</th>
<th>Max $z(A)$ – training</th>
<th>Final $z(A)$ – test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healey’s model</td>
<td>HCUI</td>
<td>3</td>
<td>0.3</td>
<td>2.23906</td>
<td>2.08418</td>
</tr>
<tr>
<td>Bingham</td>
<td>BCUI</td>
<td>5</td>
<td>0.05</td>
<td>2.25508</td>
<td>1.7615</td>
</tr>
<tr>
<td>Bingham, sq. root down-weighting</td>
<td>R2BCUI</td>
<td>5</td>
<td>0.05</td>
<td>2.23583</td>
<td>1.76098</td>
</tr>
<tr>
<td>Bingham, linear down-weighting</td>
<td>R1BCUI</td>
<td>5</td>
<td>0.05</td>
<td>2.21965</td>
<td>1.73291</td>
</tr>
<tr>
<td>Angular normal model</td>
<td>ANCUI</td>
<td>3</td>
<td>0.1</td>
<td>2.17577</td>
<td>1.61687</td>
</tr>
<tr>
<td>Gaussian oriented planar.</td>
<td>GOPCUI</td>
<td>5</td>
<td>0.1</td>
<td>2.154</td>
<td>1.55169</td>
</tr>
<tr>
<td>Gaussian normalised colour</td>
<td>GNCCUI</td>
<td>3</td>
<td>0.05</td>
<td>2.1221</td>
<td>1.50149</td>
</tr>
<tr>
<td>Trivariate Gaussian</td>
<td>3DG</td>
<td>5</td>
<td>0.05</td>
<td>2.13058</td>
<td>1.22731</td>
</tr>
<tr>
<td>Zhu and Yuille’s model.</td>
<td>ZYCUI</td>
<td>3</td>
<td>0.05</td>
<td>1.10345</td>
<td>0.9497</td>
</tr>
</tbody>
</table>

Table 11 Optimal parameter settings, maximum performance over clothes training set, and final performance over clothes test set for the ARM using various chromaticity models.
Chapter 7 Experiments and Results

The ROC curves generated over the test set for each SCM tested in this section and the last are shown in Figure 50. Performance measures of the SCMs are plotted in Figure 51, in which they are ranked in order of performance with the best model on the left.

![Figure 51 Plot of ARM performances using various statistical colour models over the clothes test set.](image)

7.4.1.2 Tracks Data Set

A similar approach is adopted over the tracks data set. Initially, the intensity models are compared using a Bingham chromaticity model. Once the best intensity model has been found in this way, it is used exclusively in a comparison of the most promising chromaticity models. Bingham chromaticity models with various down-weighting schemes, Healey’s chromaticity model and Zhu and Yuille’s model are compared. Although Zhu and Yuille’s model performs badly over the clothes data set, it is included here because it has a significantly different shape to the other directional models, which may generalise better to the less constrained environment in which the tracks data set was obtained.

At the end of the section, the performances obtained by the ARM over the test set of tracks data using each of the SCMs tested are plotted in Figure 54.

7.4.1.2.1 Intensity model comparison

Performance measures found over the tracks training set are shown in Appendix D, section D.3, and the ROC curves obtained over the test set are shown in Figure 52. The maximum
performances, corresponding optimal parameter settings and performance measures obtained over the tracks test set are shown in Table 12.

![Figure 52 ROC curves generated by the ARM over the tracks test set using various intensity models.](image)

<table>
<thead>
<tr>
<th>Intensity Model</th>
<th>SCM Abbreviation</th>
<th>$\beta$</th>
<th>$\rho$</th>
<th>Max $z(A)$ – training set</th>
<th>Final $z(A)$ – test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planar surface</td>
<td>BCPSI</td>
<td>0.05</td>
<td>3.0</td>
<td>3.39375</td>
<td>1.75285</td>
</tr>
<tr>
<td>Gaussian</td>
<td>BCGI</td>
<td>0.1</td>
<td>7.0</td>
<td>3.39912</td>
<td>1.66413</td>
</tr>
<tr>
<td>Uniform</td>
<td>BCUI</td>
<td>0.1</td>
<td>3.0</td>
<td>3.3946</td>
<td>1.44094</td>
</tr>
</tbody>
</table>

Table 12 ARM optimal parameter settings and performance measures obtained for each intensity model over the tracks training and test sets.

7.4.1.2.2 Chromaticity Model Comparison

The planar surface intensity model was found to produce the best performance over the tracks test set and so is used in the comparison of four alternative directional chromaticity models. Performances over the training set are shown in Appendix D, section D.4, the performance measures obtained using a trivariate Gaussian model are unchanged from those obtained during the ARM comparison in section 7.3.1, shown in Figure 45. ROC curves generated over the test set are shown in Figure 53 and Table 13 shows the maximum performances obtained for each intensity model.
model over the training set, the corresponding ARM parameter settings and the performance measure obtained over the test set.

![ROC curves obtained from the ARM over the tracks test set using various chromaticity models.](image)

**Table 13** Optimal parameter settings, maximum performance over tracks training set, and final performance over tracks test set for the ARM using various chromaticity models.

<table>
<thead>
<tr>
<th>Chromaticity Model</th>
<th>SCM Abbreviation</th>
<th>$\rho$</th>
<th>$\beta$</th>
<th>Max $z(A)$ – training</th>
<th>Final $z(A)$ – test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivariate Gaussian</td>
<td>3DG</td>
<td>3</td>
<td>0.1</td>
<td>3.49502</td>
<td>2.26209</td>
</tr>
<tr>
<td>Bingham, sq. root down-weighting</td>
<td>R2BCPSI</td>
<td>5</td>
<td>0.3</td>
<td>3.38673</td>
<td>1.78591</td>
</tr>
<tr>
<td>Zhu and Yuille’s model.</td>
<td>ZYCPsi</td>
<td>3</td>
<td>0.05</td>
<td>3.22951</td>
<td>1.75591</td>
</tr>
<tr>
<td>Bingham</td>
<td>BCPSI</td>
<td>3</td>
<td>0.05</td>
<td>3.39375</td>
<td>1.75285</td>
</tr>
<tr>
<td>Bingham, linear down-weighting</td>
<td>R1BCPSI</td>
<td>3</td>
<td>0.05</td>
<td>3.39271</td>
<td>1.74481</td>
</tr>
<tr>
<td>Healey’s model</td>
<td>HCPSI</td>
<td>7</td>
<td>0.05</td>
<td>3.44583</td>
<td>1.5234</td>
</tr>
</tbody>
</table>

**7.4.2 Thresholded Likelihood Images**

To reduce computation times, the use of morphological post-processing has been abandoned and the SCM comparison is made using the raw thresholded likelihood images. There are no other parameters that require tuning within this algorithm and so no training phase is required.
Chapter 7 Experiments and Results

However, so that the results are comparable with those obtained from the ARMs, performance measures are generated using just the test sets from each data set.

Figure 54 Plot of ARM performances using various statistical colour models over the tracks test set.

<table>
<thead>
<tr>
<th>Statistical Colour Model</th>
<th>Z(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Healey’s chromaticity model; uniform intensity.</td>
<td>1.29125</td>
</tr>
<tr>
<td>Bingham chromaticity, linear down-weighting; uniform intensity.</td>
<td>1.20854</td>
</tr>
<tr>
<td>Angular normal chromaticity; uniform intensity,</td>
<td>1.20452</td>
</tr>
<tr>
<td>Bingham chromaticity; uniform intensity.</td>
<td>1.20451</td>
</tr>
<tr>
<td>Bingham chromaticity, sq. root down-weighting; uniform intensity.</td>
<td>1.20060</td>
</tr>
<tr>
<td>Gaussian oriented planar chromaticity; uniform intensity.</td>
<td>1.18009</td>
</tr>
<tr>
<td>Gaussian normalised colour chromaticity; uniform intensity.</td>
<td>1.08363</td>
</tr>
<tr>
<td>Trivariate Gaussian.</td>
<td>0.97290</td>
</tr>
<tr>
<td>Bingham chromaticity; Gaussian intensity.</td>
<td>0.85064</td>
</tr>
<tr>
<td>Bingham chromaticity; planar surface intensity.</td>
<td>0.85041</td>
</tr>
<tr>
<td>Zhu and Yuille’s chromaticity; uniform intensity.</td>
<td>0.72571</td>
</tr>
</tbody>
</table>

Table 14 Performance of the thresholded likelihood image algorithm over the clothes test set using various SCMs.

The performance measures made over each test set are shown in Table 14 and Table 15 and plotted in Figure 56 and Figure 58. In those figures, the models are plotted in the same
order as in the corresponding figures produced using the ARM, Figure 51 and Figure 54 respectively. The performance still generally decreases from left to right, which shows that the models are ranked similarly for the two algorithms over both test sets. ROC curves for each SCM are shown in Figure 55 and Figure 57.

<table>
<thead>
<tr>
<th>Statistical Colour Model</th>
<th>Z(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trivariate Gaussian</td>
<td>2.26209</td>
</tr>
<tr>
<td>Bingham chromaticity, sq. root down-weighting; planar surface intensity.</td>
<td>1.78591</td>
</tr>
<tr>
<td>Bingham chromaticity, linear down-weighting; planar surface intensity.</td>
<td>1.75599</td>
</tr>
<tr>
<td>Bingham chromaticity; planar surface intensity.</td>
<td>1.75285</td>
</tr>
<tr>
<td>Bingham chromaticity; Gaussian intensity.</td>
<td>1.74481</td>
</tr>
<tr>
<td>Zhu and Yuille’s chromaticity; planar surface intensity.</td>
<td>1.66413</td>
</tr>
<tr>
<td>Healey’s chromaticity; planar surface intensity.</td>
<td>1.5234</td>
</tr>
<tr>
<td>Bingham chromaticity; uniform intensity.</td>
<td>1.44094</td>
</tr>
</tbody>
</table>

Table 15 Performance of the thresholded likelihood image algorithm over the tracks test set using various SCMs.

Figure 55 ROC curves obtained from thresholded likelihood images, with no morphological post-processing, over the clothes test set.
Figure 56 Plot of thresholded likelihood image performance over clothes test set for various SCMs.

Figure 57 ROC curves obtained from thresholded likelihood images, with no morphological post-processing, over the tracks test set.
Figure 58 Plot of thresholded likelihood image performance over tracks test set for various SCMs.
Chapter 8 Conclusions

In this chapter the experiments and results presented in the previous chapter are discussed and conclusions are drawn. The experiments and results are discussed in the order they were presented in Chapter 7, i.e., the comparison of the region tracking algorithms is discussed first, followed by the comparison of the statistical colour models. Some discussion of the methods used to assess the performance and generate comparisons is then provided.

8.1 Region Tracking Algorithms

The ranking of the performance of the three classes of region tracking algorithm deduced from the results in section 7.3 comes as no great surprise. As expected, the loose constraints on the connectivity of the region extracted by the region growing algorithm prove beneficial for the regions of interest in the two data sets used, since they are all connected, and this algorithm outperforms the simple thresholding algorithm. The smoothness and strict topological constraints inherent in the ARM algorithm, which significantly outperforms both the region growing and thresholded likelihood image algorithms, further enhance performance.

8.2 Active Region Models

The comparison of the different ARM implementations, described in section 7.3.3, produced some more surprising results. As expected, performance becomes more reliable if the CID (collision implies deletion) strategy is adopted rather than the OIP (original Ivins and Porrill) move reversal strategy when two separate parts of the contour come into contact. Use of the CID strategy avoids the tangling effect, illustrated in Figure 27, section 5.3.2, which can occur with the OIP and cause unusually low performance measures for some combinations of parameter settings. Examination of the performance over the training sets, shown in Figure 45 in section 7.3.3, highlights this phenomenon. At most combinations of parameter settings, the performance of the OIP and CID ARMs, with or without the termination condition, is very similar as few collisions occur and both models successfully resolve those that do. Occasionally, however, collisions do occur and are not successfully resolved by the OIP ARM causing overall performance to suffer. One combination of parameter settings for which difference in performance is particularly apparent is $p=3.0, \beta=0.3$. When the termination condition is not used, this combination is found to be optimal for the CID ARM but produces very poor performance in the OIP. With these parameter settings, tangles occur in the OIP ARM at several of the low operating points when it is applied to the first training sequence (top left in Figure 40) causing the false positive rate to be unexpectedly high. Figure 59 shows the final configurations of both the OIP and CID ARMs, at one of these operating points, on some of the
Chapter 8 Conclusions

images of this sequence. The images shown are the 4th, 5th, 12th and 13th frames of the sequence. As highlighted in the figure, the OIP ARM becomes tangled on the 5th frame and this tangle remains until the 12th frame but is finally resolved on the 13th. The resulting false positive count is higher than for the CID ARM, which does not get tangled, even though the true positive count is similar and so the measure of overall performance is decreased.

Figure 59 An example illustrating how low measures of performance arise for the OIP ARM (shown in purple), which is compared to the CID ARM (shown in blue).
Chapter 8 Conclusions

Also as expected, the DAG (discrete averaged goodness) ARM produces higher performance than the CID ARM. As discussed in 5.3.5, if the goodness of the average pixel value over a neighbourhood of each contour element is taken rather than the average of the goodness, some neighbourhoods of pixels with individually low goodesses can have an average pixel value with deceptively high goodness. For a fixed true positive ratio, this effect will tend to increase the false positive ratio and thus bring down the overall performance.

One further point to note concerning the performance of the variational ARMs is that the use of the termination condition proves consistently to improve overall performance. The main purpose of the development of a termination condition was simply to reduce overall computation time and the profound effect on overall classification performance is a somewhat surprising bonus. Once the vicinity of the true boundary has been found the value of the AFM (average force magnitude) that is thresholded for termination varies noisily about some stable value, see Figure 29 and Figure 30, in section 5.3.4.1. The magnitude of the force applied to a contour element is only an estimate of the change in energy will occur as a result of the application of that force to that element. Even to find that the contour configuration for which the AFM was smallest over a fixed number of iterations is consistently closer to the ground truth than the effectively randomly selected configuration after the last iteration is somewhat surprising. However, the termination condition used here actually selects the subsequent configuration, as the AFM is calculated during a full iteration in which the position of each contour element is updated. As mentioned in section 5.3.4, see Figure 28, once an image boundary has been located by a variational ARM, the position of an individual contour element can vary considerably and may cover a range of several pixels perpendicular to the true image boundary. The contour after the one with the minimum energy configuration will still have similar configuration, as each element can have moved only one pixel at most. Thus, if the minimum energy contour is assumed to be the closest approximation to the ground truth, the contour selected should also to be a better than average approximation. The results of the comparison in section 7.3.3 give weight to this argument and strengthen the case for convergence of ARMs being a desirable property.

It is difficult to find examples to illustrate the effects discussed in the previous paragraph, because the introduction of the termination condition often causes both the false positive and true positive ratios at a particular operating point to change significantly. However, one operating point was found to produce a similar number of false positives for the DAG ARM both with and without the termination condition, but a higher true positive count when the termination condition was applied. The final configurations of the DAG ARM with and without the termination condition on each of the first four images of the sequence are shown in Figure 60. In this example, the operating point corresponds to a low value of k, which gives rise to the conservative regions segmentations observed in the figure. It appears that the differences
between the contour are more local to particular groups of elements than would be suggested by
the explanation given in the previous paragraph. When the termination condition is used, parts
of the contour are not allowed to contract as far and so the true positive count remains higher.
However, it seems reasonable that this effect could work both ways, increasing the false
positive count at higher operating points. Small differences in other parts of the contour can be
observed, but it is difficult to decide whether one contour is better than the other in these
regions.

Figure 60 Evolution of a DAG ARM, over the first four frames of a sequence – from left to right,
top to bottom, with (in red) and without (in blue) the termination condition applied.

Perhaps the most surprising result obtained from the comparison of the ARMs is the
performance of the different greedy ARMs. Despite the findings discussed in the last paragraph,
the overall performance of these ARMs decreases noticeably with increased stability and
tendency to converge. Thus, the SIG (surface integral greedy) ARM, which is almost guaranteed
to converge, produces worse results than the less stable MLG (multi-line greedy) ARM, which
performs worse than the even less stable SLG (single line greedy) ARM. Furthermore, all of the
greedy ARMs produce lower overall performance than the highly unstable variational DAG ARM.
The problem with the more stable models appears to be that they are more susceptible to
weak local minima in the image. As soon as the SIG ARM detects even a very slight minimum
point in the energy surface, no further movement in the corresponding part of the contour will
occur. The instability of the DAG ARM, however, ensures that even when the contour
configuration corresponds to fairly strong energy minima, subsequent movement is still assured.
This allows the DAG ARM to pass over weak minima with ease, but also causes chaotic
oscillations in the vicinity of the global minimum. The MLG ARM, and even the SLG ARM, are also susceptible to such local minima, but to a lesser extent than the SIG, and this shows in the performance measures taken from these models. An example of a greedy ARM, in fact a MLG ARM, failing to overcome a local minimum that is passed over by the DAG ARM is shown in Figure 61. The value of $k$ used for the MLG ARM in the figure is 3.26, whilst that for the DAG ARM is 3.14. Accordingly, at most positions around the contour, the MLG contour is outside that of the DAG. However, despite the higher level of tolerance, the MLG ARM remains inside the DAG ARM at one position on the right hand side of the track and excludes some pixels that are classified positively in the ground truth image.

Figure 61 An example of a MLG ARM (shown in purple) getting stuck at a local energy minimum position, which is passed over by the DAG ARM (shown in blue).

One further point that should be made concerning the greedy ARMs is that their behaviour has been less extensively investigated than the variational models. It appears from the training phase, see Figure 45 and Table 7 in section 7.3.3, that the optimal settings for the parameters are slightly higher than for the variational models. For the MLG ARM, the maximum point appears well within the range of settings of the parameters but for the SLG, the optimal settings are at the limit of this range. In fact, examination of the performance with a slightly higher value for the image energy coefficient does establish this combination of settings as at least a local maximum, but it is possible that the optimal parameter settings could lie well beyond the range investigated. Due to the constraints of time, no training phase has been performed for the SIG ARM and so the test results are not fully conclusive.

Susceptibility to tangling, or local minima, are both artefacts of the ARM algorithm that can decrease the measured performance independently of the choice of underlying statistical colour model. Similarly, the introduction of extra false positives caused by the incorrect neighbourhood averaging method used in the OIP and CID ARMs causes the performance measure to decrease because of deficiencies in the algorithm rather than the underlying statistical model. Also, the random inclusion/exclusion of pixels close to the true boundary of the region of interest that can occur when variational ARMs are terminated after a fixed number of iterations further increases noise in the performance measure that is independent of the choice.
Chapter 8 Conclusions

of underlying model. As can be seen from the slightly noisy variation in the performance measures obtained over the tracks training set, shown in Figure 45, all of the ARM implementations suffer from some of these deficiencies to a greater or lesser extent. The choice of the best ARM with which to compare the underlying SCMs, as mentioned in section 7.4.1, should be the one that is least affected by these artefacts and not necessarily the one that produces the highest performance in the comparison of section 7.3.3. However, as all of these artefacts have a tendency to decrease the measure of performance, it seems reasonable, in the absence of any other clear indications, to assume that the ARM with the highest performance is the ARM that is least prone to these detrimental effects. The DAG ARM with the termination condition was found to produce the highest overall classification performance of all the ARMs (and indeed region tracking algorithms) tested and so was selected for use in the statistical colour model comparison.

8.2.1 General Conclusions

Although some improvements have been made over the original ARM, problems remain. The ARM algorithms still contain artefacts that can make their performance difficult to characterise. Although the tangling problem, illustrated in Figure 27, section 5.3.2, is largely overcome by the collision implies deletion strategy, resolution of tangles can be slow and occasionally the true boundary may not have been found by the time the maximum number of iterations has been performed. This can be seen in Figure 27 where almost 200 iterations are required to find the true boundary even on this very simple image.

Artefacts in the algorithm cause the performance to vary in an unpredictable way as the internal parameter settings are varied. This unpredictability can be observed in the performance measures taken from ARMs over the training sets, as the stiffness and image energy coefficients are varied, see Figure 45 and Appendix D. In general, the measure of performance does not vary smoothly as one parameter is varied and unexpectedly poor performance is occasionally obtained from a particular combination of settings. When applying the performance characterisation methodology, as a result of this unpredictability, the optimal parameter settings selected for different SCMs varies quite considerably, even over the same training set, see Table 11 and Table 13 in section 7.4. As the optimal parameter settings affect only the shape of the extracted region, they might reasonably be expected to be independent of the underlying statistical colour model and should be consistent over the same training set. One possible source of the observed noise in the performance measure is the fixed topology of the ARM. Figure 62 shows a situation where this restriction introduces some noise even for a very stable (MLG) ARM as a fixed position can never be found. The value of k is quite small and negative goodness is assigned to some dark portions of the region of interest (the red garment). The contour tries to exclude two such regions, one at the top right of the region of interest and one at

175
the bottom left. However, each of these regions are surrounded by pixels with positive goodness, which the contour tries to include, and the result is that the contour tries to surround these dark regions. Two separate parts of the boundary thus collide in both these regions causing contour elements to be repeatedly added and deleted. This situation is never resolved and eventually the maximum number of iterations is reached so that the final configuration is somewhat random.

Figure 62 An unstable final configuration caused by the fixed topology of the ARM (shown in blue).

Another possible source of noise in the performance measure is the method used to extract a region segmentation from the final contour position found by an ARM. As mentioned in 5.3, from a discretely defined contour a binary region segmentation is extracted using a raster scan line polygon filling algorithm that excludes every pixel that lies on the actual boundary. This method has problems, for example, as can be seen in the typical track region segmentation obtained in this way shown in Figure 26, section 5.3, pixels along the edge of the image are excluded even when the contour lies along that edge and cannot move out any further. The number of pixels lying on a typical ARM contour is roughly $1/1000^\text{th}$ of those contained in the entire image and thus an error of this same size is potentially introduced into the measure of the area under an ROC curve produced by an ARM. Such errors however are highly correlated when comparisons are being made and so are assumed to have negligible effect on the results of the comparisons in the previous chapter. A better approach might be to exclude pixels that lie on the actual contour from the classification entirely, effectively introducing a doubt category into the region segmentation produced by the algorithm.

8.3 Statistical Colour Models

In this section some conclusions are drawn from the experimental comparison of the SCMs presented in section 7.4. As mentioned in that section, the results obtained from the two separate comparisons, obtained by assessment of two different classification algorithms, were very similar. Neither algorithm is ideal for performing this comparison. The ARM algorithm is a complex dynamical system and is prone to introducing artefacts of its own into the performance measures taken from it to represent the classification power of the underlying SCM. The
thresholded likelihood image algorithm introduces no artefacts of its own, but does not exploit
available spatial knowledge and allows misclassifications to occur well away from the boundary
of the region of interest. This renders performance measures taken from the algorithm more
susceptible to non-uniformity of the background region in the image and more constrained
models may produce better performance than ones that are more consistent with the parent
distribution.

8.3.1 Clothes Data Set

From the ROC curves shown in Figure 49, the uniform intensity model can clearly be seen to be
the best choice for use with the ARM over the clothes data set. The corresponding ROC curves
for the thresholded likelihood image algorithm using a Bingham chromaticity model with each
of the three intensity models, shown in Figure 55, confirm that the same applies for this
algorithm. The Gaussian and planar intensity models have very similar performance in both
cases, particularly for the thresholded likelihood image algorithm where the ROC curves appear
identical in Figure 55, but the Gaussian intensity has a slightly larger $z(A)$, see Table 11. For the
ARM, different pairs of optimal parameter settings were found for these two models during the
training phase, which causes a larger difference between the measured performances of the two
models over the test set. Likelihood images obtained, from the shorts image of Figure 3, using
uniform and Gaussian intensity models, with a Bingham chromaticity model, and moderate
values of $k$ are shown in Figure 63. In that figure the greater degree of discrimination provided
by the uniform model is clearly visible. The region of interest is generally lighter and the
background, apart from a few patches, is generally darker. The likelihood images obtained using
the planar and Gaussian intensity models appear identical and so only one is shown.

With the uniform intensity model, all of the directional chromaticity models, presented
in section 3.3, are subsequently compared in section 7.4. In general, the hypotheses of section
3.3 are upheld by the results obtained during that comparison. The tighter models obtained by
representing chromaticity as a point in a plane perpendicular to the mean of a single mode
chromaticity distribution, rather than in a fixed plane such as the normalised colour space, is
found to improve performance. Likelihood images generated using both of these models are
shown in Figure 64. Both these images appear similar and it is not clear from them that one
model provides better discrimination than another does.

Slight differences can however be observed (at least if viewed consecutively to
highlight differences) between the images in Figure 64 and the likelihood image from the
Bingham chromaticity model with uniform intensity shown in Figure 63. In all three likelihood
images, the region of interest appears similar, but some parts of the background are slightly
darker in the Bingham likelihood image, for example the dark region on the far left of the
image. All of the spherical models – the Bingham models, the angular normal model and
Healey's model, produce higher measures of performance than the planar models, when used by either algorithm. Neither of the down-weighting schemes, used to decrease the effects of noisy data near the origin of the RGB cube, as discussed in 3.3.6, appears to have a significant effect on the performance of the Bingham model. Likelihood images for the three versions of the Bingham model appear identical.

Figure 63 Likelihood images obtained, from the shorts image of Figure 3, using uniform (left) and Gaussian (right) intensity models and a directional (Bingham) chromaticity model.

Figure 64 Likelihood images obtained, from the shorts image of Figure 3, using Gaussian chromaticity model in a fixed plane (left) and an oriented plane (right).

Figure 65 Likelihood images obtained, from the shorts image of Figure 3, using Healey's model (left) and an angular normal model (right).

Despite the problems discussed in section 3.3.4, the angular normal model performs well and has performance comparable to the Bingham models when used by the thresholded likelihood image algorithm, though it is noticeably worse when used by the ARM. As the
distributions are both tight about their mean direction and not in the vicinity of the pole of the sphere, the shape of the distribution is virtually symmetric, as can be seen from Figure 12, in section 3.3.4, and so a good model is provided. A likelihood image generated from an angular normal model is shown in Figure 65 and appears very similar to those generated from the Bingham models.

The best performance over the clothes data set, for either algorithm, as can be seen directly from the ROC curves in Figure 49 and Figure 55, is obtained from Healey's chromaticity model. The picture of the model in Figure 16, section 3.3.7, shows that this chromaticity model is a much tighter fit to the sample distribution of the green shorts than any of the other directional models. Likelihood images constructed from a Healey chromaticity model and an angular normal model are shown in Figure 65. The image from Healey's model is noticeably darker at the extremes of the image than those from the other spherical models, for example the angular normal model whose likelihood image is shown alongside, and the region of interest is slightly lighter.

Finally, the other two models compared in section 7.4 are the trivariate Gaussian model and Zhu and Yuille's model. Likelihood images from each of these models are shown in Figure 66. It is clear from the ROC curves obtained from each algorithm, see Figure 49 and Figure 55, that the Gaussian model does not perform as well as the directional models. The likelihood image in Figure 66, suggests that a lesser degree of discrimination is provided by this model than the directional ones discussed above. These results indicate that the shape of the trivariate Gaussian distribution is not as good an approximation to the shape of the parent distributions in the clothes data set as the conical, directional chromaticity, uniform intensity models.

The worst performance is obtained from Zhu and Yuille’s model. The likelihood image in Figure 66 clearly shows that a high false positive count can be expected even when the true positive count is low, as high likelihood is assigned to all the pixels in the dark regions at the outskirts of the image. This is confirmed by the ROC curves obtained from this model using either algorithm. Despite the overall poor performance obtained from this model, for very high operating points with the thresholded likelihood image algorithm, it does appear to have some advantage. As can be seen from the ROC curves in Figure 55, for very high true positive ratios, Zhu and Yuille’s model has the lowest false positive rating. In the likelihood images of the other directional models, although the dark pixels in the outskirts of the image are assigned much lower likelihood, so are the dark pixels in the region of interest. Because the shape of Zhu and Yuille’s model is cylindrical rather than conical, these dark pixels are assigned reasonably high likelihood even at much lower operating points. The same is true, but to a lesser extent, of the trivariate Gaussian model, which also does not have such a severe point at the origin of the RGB cube and so assigns higher likelihood to dark pixels whose chromaticity might be very
Chapter 8 Conclusions

different to the mean. From Figure 55, it can be seen that for the trivariate Gaussian also, the maximum true positive ratio is reached before the conical models.

In the comparison made using the ARM, this effect does not appear. The spatial constraints in the algorithm ensure that the dark pixels in the region of interest are positively classified despite having low likelihood.

![Likelihood images obtained, from the shorts image of Figure 3, using a trivariate Gaussian model (left) and a Zhu and Yuille SCM (right).](image)

8.3.2 Tracks Data Set

It is not clear from the derivation of the planar surface intensity model, in section 3.4.3, whether or not this model is applicable in daylight. The derivation relies on a very simple reflection model and assumes that the incident light is roughly collimated. As discussed in section 2.1.5, there are many effects that are not considered, such as local or global vignetting and inter-reflection, which may corrupt this model. However, the ROC curves in Figure 52 show that, in the comparison performed with the ARM, slightly better performance is obtained using the planar intensity model than the Gaussian and so it is adopted in the subsequent comparison of the chromaticity models. The planar surface intensity model is also found to outperform the Gaussian with the thresholded likelihood images, but the margin between the models is narrower. In both cases, the uniform intensity model does not perform as well. Little difference between the likelihood images generated using the Gaussian and planar surface intensity models can be observed, but those from the planar and uniform intensity models, both combined with a Bingham chromaticity model are shown in Figure 67. Use of the more restricted intensity model can clearly be seen to reduce the likelihood applied to several background regions, in particular the sky, while mostly maintaining the high likelihood assigned to pixels in the region of interest.

Three models of similar shape are compared to the Bingham chromaticity model with the planar surface intensity model – Healey’s chromaticity model and the two robust Bingham models, all with the same intensity model. Over this data set, Healey’s chromaticity model is not as effective and its performance is significantly worse than the Bingham based models. The effect of the down-weighting schemes on the Bingham model is once again not fully clear. They
prove to be an asset and increase overall performance when the thresholded likelihood image algorithm is used, but seem to have little effect when used with the ARMs. This suggests that only the likelihood of some pixels well away from the region of interest are affected, but no example has been found. Likelihood images are shown for Healey’s chromaticity model and a linearly down-weighted Bingham chromaticity model, both using a planar surface intensity model, in Figure 68. The down-weighted Bingham image appears identical to the standard
Chapter 8 Conclusions

Bingham image in Figure 67, but the likelihood image created using Healey’s chromaticity model is noticeably lighter in some background parts of the image.

The other two models compared over this data set are the trivariate Gaussian model and Zhu and Yuille’s chromaticity model, again using a planar intensity model. Likelihood images generated using these models are shown in Figure 69. The Gaussian model is found to outperform all the directional models by some margin using either algorithm, as can clearly be seen from the ROC curves in Figure 53 and Figure 57. As mentioned in section 3.4.2, when Zhu and Yuille’s chromaticity model is combined with a Gaussian intensity model, the full SCM obtained is similar to the trivariate Gaussian model. The main difference is that the trivariate Gaussian model is not constrained to have one of its axes lie along a line passing through the origin of the RGB cube. As the planar intensity model has proved to be a slight improvement over the Gaussian intensity model, it is reasonable to expect that the fusion of this intensity model with Zhu and Yuille’s chromaticity model might yield a full SCM that is a slight improvement over the trivariate Gaussian SCM. However, this does not prove to be the case and the overall performance measure obtained from this model is similar to that of the Bingham (although the behaviour at particular operating points is quite different as can be seen from the ROC curves, in particular in Figure 57). It appears then that the greater orientability of the Gaussian model is a considerable advantage in the outdoor environment in which the tracks data set was obtained. Although the likelihood image from Zhu and Yuille’s chromaticity model is noticeably different to those from the Bingham distributions, it is hard to decide which provides the highest degree of discrimination of the region of interest from the background. The likelihood image from the Gaussian model, however, is noticeably better with several background regions appearing darker – again the sky is a good example.

8.3.3 General Conclusions

Some general conclusions can be drawn from the observations and discussion made above. Intuitively it seems that the sphere is a more appropriate domain for representation of directional data than the plane and this hypothesis has been verified at least for the directional chromaticity data under investigation in this thesis. The issue of very dark pixels, for which the chromaticity is poorly defined, has proved to be an important one. In the clothes data set, a significant proportion of the pixels in each image comes from this region of the RGB colour cube. Most of those pixels are not part of the region of interest and so assigning high likelihood to them, as in Zhu and Yuille’s SCM, greatly increases the false positive count and brings down overall performance of the model. However, when low likelihood is assigned to these pixels, as in the conical SCMs, higher false negatives rates can be expected. In the clothes data set, the region of interest is small compared to the rest of the image and so less misclassifications occur overall if
dark pixels are assigned low likelihood, however, in general it is not clear how a balance between these two effects should be struck.

One thing that is clear is that dark pixels, with unreliable chromaticity, in the sample distribution should not be allowed to have a large influence on the shape of the model. The down-weighting schemes used to decrease this effect have proved to have only a minor effect on the shape of the model, but can increase performance slightly. As illustrated in Chapter 3 and reinforced by the experimental findings in the last chapter, Healey’s chromaticity model avoids the effects of noisy dark sample data more effectively by extracting the statistics of the data before normalising to obtain a directional model.

When the data is less noisy and mainly resides in regions of the RGB colour cube where the chromaticity can be relied upon, Healey’s directional model is less effective and better performance is obtained from the Bingham model. This is illustrated by the comparison of the directional models over the tracks test set.

In general however, the directional models do not generalise well to environments where the physical model used to derive them is less applicable. As discussed at the end of the last section, the orientability of the Gaussian model allows a tighter fit to the sample distribution and higher classification rates over the parent distribution to be obtained. The Gaussian model in fact provides the best performance over all but one of the individual track sequences – a BCUI model is best over the sequence used in the example shown in Figure 61, in which there is little illumination directly from the sun. Although it generalises more readily than the directional models, there is little physical justification for the use of the Gaussian model in these situations. In section 3.6.3 it was suggested that, in the absence of directional light in outdoor scenes so that the only illumination comes from the sky, the Gaussian model might be appropriate, but in many of the images in the track data set, there is considerable illumination from the sun.

8.4 Performance Characterisation Methodology

The methodology used to characterise the performance of the region tracking algorithms used in this work has proved to be effective both for comparing the performance of different algorithms and as a device for comparing the quality of underlying models used by those algorithms. The use of ROC analysis is appealing because it allows comparisons to be made between any binary classification algorithms and any underlying model that can be used by them. All of the algorithms used in this work can be regarded as classification algorithms and the ability to obtain quantitative measures of their performance, afforded by the performance characterisation methodology, has proved invaluable.

183
Chapter 8 Conclusions

For the most part it is impossible to make definitive decisions by inspection of the output of algorithms about which algorithm or underlying model provides the best performance. Hypotheses can be made by inspection, however, and many of those that were made in the preceding chapters have been quantitatively confirmed by use of the experimental protocol of Chapter 6. For example, the superiority of spherical directional models over planar ones; and the higher performance of the more constrained ARM region tracking algorithm over simpler region growing and thresholding techniques. In the pictures of the chromaticity models given in Chapter 3, the models based on spherical representations of chromaticity appear to fit sample and parent distributions more tightly and so it is reasonable to expect that they classify more accurately. However, when likelihood images generated from these models, some examples of which are shown in the previous section, are inspected it is very hard to rank the degrees of discrimination that is provided by each. Similarly, by inspection of the results obtained from an ARM using various underlying models on several images, ranking the performance of the underlying models is extremely difficult, not least because one can never be sure if better results might be obtained with different internal parameter settings.

Sometimes, very strong hypotheses can be made by inspection however, for example, that an ARM will produce better rates of classification than a simple thresholding technique when the region of interest is known to be connected and compact. Other hypotheses, such as that the DAG ARM should provide better classification performance than the CID ARM, can be drawn from theoretical considerations but again are difficult to confirm by inspection. It is encouraging that the quantitative results obtained confirm such intuition and these findings allow more trust to be placed in more surprising results.

Another advantage of the use of a performance characterisation methodology such as the one used here is that the behaviour of algorithms is investigated over a wide range of parameter settings and also over a wide range of imagery. This can highlight problems with algorithms that might have remained undiscovered if the behaviour were only investigated over a few images with idealised internal parameter settings. An example is the tangling problem of the original Ivins and Porrill ARM. Although this problem is unlikely to arise if the internal parameters of the ARM are tuned for a particular task, when the parameter settings are less than ideal, as they might be in an unconstrained real world application, this problem may often occur.

However, there are some problems with both the overall methodology and the actual performance measure. In the overall methodology, no tests are performed on the training and test sets to check that they are sufficiently extensive samples of the image ensembles they represent. In particular no validation set is used. Comparative results have proved to be reasonably consistent between the training and test sets, but there have been some discrepancies, for example the MLG ARM performs much better over the tracks training set than the SLG,
which significantly outperforms it over the test set. Furthermore, all of the performance measures obtained over either test set are significantly lower than the corresponding measures obtained over the training set. This is to be expected as the optimisation is performed over the training set, but the difference is quite large and indicates that more consistent and reliable results might be obtained if the training sets were extended. This may also be a contributory factor in the considerable variation that occurs in the optimal ARM parameter values that are selected for different SCMs.

Computation of individual ROC curves from which the actual measure of performance is derived can also be troublesome. The selection of an appropriate set of operating points can be difficult, as behaviour at a particular point can vary considerably from image to image. Appropriate values must be determined to ensure that the true ROC curve is sampled evenly so that the behaviour of the algorithm is properly characterised. Appropriate values may vary between the training set, over which they are selected, and the test set. For example, consider the ROC curves generated for various SCMs over the tracks test set, using the ARM, shown in Figure 53, section 7.4.1.2.2. Although the superiority of the Gaussian model and the inferiority of Healey's chromaticity model are clearly evident from this graph, it is hard to rank the performance of the other four models. Many of the operating points appear to be set too low and the portion of the curve where the behaviour of the models differs is poorly sampled. Despite this uncertainty, the models are ranked quite definitively by the performance measure, z(A), as shown in Table 13, in the same section. Even when the uncertainty in z(A), discussed in section 7.2.4, is taken into account, this ranking is quite conclusive – at the levels of performance indicated by these curves, an error of $10^{-4}$ in the area under the curves corresponds to an error of about $1.5\times10^{-4}$ in z(A). This suggests that the estimate of the uncertainty in z(A) may be too low.

A limitation of the applicability of the ROC performance measure is that there must be a parameter, such as the SCM tolerance parameter, $k$, contained in the algorithm that directly effects the true and false positive counts and so can be used to parametrise ROC curves. For algorithms that do not exploit statistical models, for example and edge based ACM (active contour model), parametrisation of the ROC curve may be harder.
Chapter 9 Achievements and Future Work

In the first chapter of this thesis, the objective of this work was described as an investigation into how best to exploit colour data for applications such as tracking regions of interest through sequences of images. Three questions to be addressed were posed and in the body of the thesis answers to those questions have been sought. Some success has been achieved in each of these three areas and in this final chapter, these achievements are summarised and possible directions for future work are discussed.

9.1 Colour Modelling

The first question was concerned with how best to characterise objects in terms of their colour data and was addressed in Chapter 2 and Chapter 3.

9.1.1 Achievements

In Chapter 2, physical models of the processes that give rise to distributions of colour data were investigated and the techniques available for modelling these distributions were reviewed. Some success in applications similar to those motivating this work had been reported when parametric statistical models were used to characterise the distribution of colour data arising from an object of interest. The investigation was thus restricted to the problem of determining suitable parametric statistical models for these distributions. In Chapter 3, parametric statistical models that have been used for modelling distributions of colour data in the past were examined in detail and their properties compared to each other and to some new alternatives based on more standard directional statistical models. As well as models for the chromaticity component of colour data, some discussion of the intensity component is also provided. A novel statistical model for the distribution of intensities corresponding to planar objects is proposed. New statistical colour models that combine directional models of the chromaticity component of the data with models for the intensity are constructed.

In Chapter 7, an empirical comparison of these models was performed over image data representative of two separate applications. This comparison revealed that the commonly used directional models for colour data perform well under ideal imaging conditions but generalise poorly to less constrained environments even when combined with appropriate non-uniform intensity models. It was further found that modelling chromaticity distributions on a sphere centred at the origin provides better models than when chromaticity is represented by coordinates in a plane, for example, as in commonly used representations for colour data such as the CIE chromaticity plane or the normalised colour space.
Chapter 9 Achievements and Future Work

9.1.2 Further Work

There are three main areas in which further work on statistical modelling of distributions of colour data might proceed – refinement of the shape of single mode models so that a tighter fit to real distributions can be obtained; generalisation of the models to multimodality; and incorporation of background models.

9.1.2.1 Single Mode Models

All of the statistical models discussed and developed in Chapter 3, are based on a very simple, unichromatic, model of the reflection process. In Chapter 2, factors such as surface reflection and multiple illuminations were discussed that can corrupt this simple model and affect the shape of the resulting distribution of colour data. The simple models based on the unichromatic model were found to perform badly when the effects of some of these factors are non-negligible, as in the tracks data set. In section 2.1.5, it was suggested that the incorporation of an ambient lighting term into the reflection model might provide a more appropriate model for daylight scenes. Although consideration of effects such as global vignetting have lead to some condemnation of the ambient lighting model, [KV96], these effects are often small on the planar surfaces of the objects of interest in the outdoor scenes of interest here. It is, therefore, reasonable to expect that the spectral composition of the light reflected from the sky is roughly constant.

Incorporation of this term into the reflection model has the effect of shifting the focal point, from which directions are measured to represent chromaticity, away from the origin, and also introducing some uncertainty in its position. In section 8.3.2, it was concluded that the reason for the poor generalisation of the directional models to data obtained from daylight illumination was that the models could not be oriented so that the major axis lies along a line that does not pass through the origin. The trivariate Gaussian model does possess this ability and performs much better in daylight environments. Other than the large number of contributory factors that combine to give real colour distributions their shape, however, there is little physical justification for use of a Gaussian and better models may be possible to obtain by using directional models that are appropriately offset from the origin. The trivariate Gaussian model itself provides an estimate of the axis along which the focal point for the distribution should lie and the actual position along that line could be estimated from sample data. A maximum likelihood estimate of this position could be made either analytically or by using a method similar to that used to estimate maxI in section 3.4.3, see Appendix B, section B.1.

There are many other factors that could be included in the reflection model on which the generic shape of a statistical colour model is based, such as surface reflection, local and global vignetting, multiple illuminations and inter-reflection. Often these factors will be negligible and their influence on the generic model shape may detract from overall performance.
Chapter 9 Achievements and Future Work

by causing over-fitting to sample data. In some cases, however, some or all of these factors may have a large influence on the shape of the true distribution and so performance could be enhanced by their consideration. There is much further work that could be done to generate more specific statistical colour models for applications where these factors are strong.

9.1.2.2 Multi-modality

In Appendix A, a brief investigation into the construction of multimode SCMs is presented. Such models could be used to characterise clustered distributions of data that might be obtained from multicoloured objects, or objects under multiple illuminations. The preliminary results that are presented in that appendix show poor performance of these models compared to single mode models even when the object of interest is clearly multicoloured. There is further work to be performed here also and the use of more constrained clustering procedures, such as a k-means algorithm, perhaps together with information criteria to select the most appropriate number of clusters, as in the work of Zhang and Modestino, [ZM90], might provide better results.

9.1.2.3 Background Models

The performance of classification algorithms may be enhanced by the use of a model of the background, as in the works of Crisman, [Cri92], and Thorpe, et al, [THK88]. Throughout this thesis, a uniform model of the colours comprising background regions of the image has been implicitly assumed, but often this distribution of colours can be highly non-uniform particularly in restricted image domains when colour variation is small. The use of background models may also help alleviate the problem of dark pixels, discussed in section 8.3.3. The chromaticity of dark pixels is poorly defined and so it is difficult to make a classification for these pixels based on colour information alone. The decision of how to assign likelihood to dark pixels can have a dramatic effect on the performance of a model, as illustrated by the very poor performance of Zhu and Yuille's model over the clothes data set. Use of a background model would provide some information about the relative numbers of dark pixels contained in each region and so give some indication of the likelihood that should be assigned.

9.2 Region Tracking

In Chapter 4 and Chapter 5, the second question was addressed – how best to exploit a model of the colour data corresponding to an object of interest for the purposes of tracking that object through sequences of images.

9.2.1 Achievements

Chapter 4 contains a review of techniques available for performing region tracking tasks. Attention is focussed mainly on techniques based on the active contour model, on which a
considerable amount of work has been performed in recent years with successful results often reported. A variation of the active contour model, the ARM (active region model), is based on an underlying statistical model of the image data and so can be used in conjunction with the statistical colour models discussed previously.

In Chapter 5, an existing ARM, developed originally by Ivins and Porrill, is taken as a starting point. Problems with this algorithm are identified and to some extent ameliorated. Several novel implementations of the ARM are suggested, in particular using a greedy approach to the energy minimisation procedure that is central to the algorithm. Such approaches have become popular for use with ACMs (active contour models) because of the greater stability and computational efficiency they provide over other approaches such as the original variational method. By approximating the energy change caused by a move of a single element of the contour using the surface integral of a bilinearly interpolated goodness function over the image, full convergence is almost guaranteed for one of the greedy implementations.

The performance of the ARMs are compared with each other and with older more established techniques for extracting a region of interest based on region growing and thresholding methods. Results show better performance from the ARM, which exploits the connected and compact nature of the regions of interest, both in terms of computation time and overall classification.

Some surprising results are obtained when the classification performances of the different implementations of the ARM are compared. The more stable ARMs based on the greedy energy minimisation scheme prove to be more susceptible to local minima in the energy landscape and thus produce worse performance overall. Mainly due to their instability, variational ARMs are able to jump out of shallow local minima and continue deformation towards deeper global minima. This instability, however, means that oscillations occur in the contour once the vicinity of the minimum energy configuration has been found and so the approximation to the true minimum configuration is noisy. The use of a termination condition that identifies configurations of lower than average energy is shown to improve overall classification performance of the variational ARMs.

9.2.2 Further Work

From the review provided in Chapter 4 it is clear that there are many ways in which the ARM algorithm could be modified. For the most part, due to the general lack of performance characterisation of these models, one cannot be certain of the ways in which these modifications affect the overall performance of the model. Comparison of any of the available implementational options with existing models would doubtless be useful, but of particular interest might be an investigation of the performance of ARMs using alternative contour representations such as B-splines. For some specific applications, the incorporation of shape
constraints and dynamic models into the algorithm may also enhance performance. All of these considerations, as well as enhancing overall performance, can help to reduce computation times and might assist in the construction of real-time implementations of ARMs. Computational complexity has been a minor consideration in this project but for the applications motivating the work, real-time algorithms would be required and this is thus an important consideration for the future.

The performance of the ARMs has been far from fully characterised. There are other internal parameters of the algorithm, for example those controlling the number of elements in the contour, whose effects have not been investigated extensively. The effects on performance of varying these parameters would be interesting to characterise. A more extensive analysis of the behaviour of the models at other combinations of the energy coefficients would also be valuable, particularly for the greedy models, which have not been investigated over a wide range of different settings.

The results of the ARM comparison show both that stability leads to susceptibility to local minima, but that the use of a termination condition that identifies low energy configurations, such as that proposed in section 5.3.4.1, can improve overall classification results. As discussed in section 8.2, that termination condition has limitations and does not actually identify the ARM configuration with the minimum energy. Further investigation into termination conditions for variational ARMs could be valuable, but the results suggest more strongly that, for optimal results, the use of a multistage energy minimisation process, such as that used by Wang, et al, [WEH96], or suggested by Ivins, [Ivi96], might be beneficial. Once in the vicinity of the global minimum energy configuration, the SIG (Surface Integral Greedy) ARM, which is guaranteed to converge, should find a lower energy configuration than any variational ARM and so produce better results. If this ARM were used to refine the final configuration of, say, the DAG (Discrete Averaged Goodness) ARM, which was found to provide the best classification when initial configuration can be far from the desired boundary, overall classification performance might improve.

A drawback of the SIG ARM is the heavy computational load it imparts. However, as mentioned in section 5.3.6.1.3, the use of the bilinearly interpolated goodness function is not necessary to ensure convergence, and estimation of the energy using the surface integral of the discrete goodness function derived directly from the discrete image data could be used instead. A comparison of the performance of these two models would be interesting but it seems reasonable to expect that performance would be very similar. Since these surface integrals would be much simpler to calculate, a speed up of around 5 or 6 times might reasonably be expected, and a model implemented in this way could make the multistage algorithm proposed in the last paragraph much more practical.
One further area that might warrant attention in the future is the further minimisation of artefacts in the algorithm that cause performance measures to be noisy. As mentioned in section 8.2.1, the topological restrictions on the ARM can cause unpredictable behaviour. One possibility for reducing the levels of noise might then be to relax this restriction, either by representation of the contour as a propagating front, see section 4.4.5.3, or by use of a scheme such as that of McInerney and Terzopoulos, [MT95], to allow contours to split and merge. Note that the use of such a scheme should further improve the convergence properties of the SIG ARM, though convergence would still not be completely guaranteed.

9.3 Performance Characterisation

The final question posed in the first chapter was concerned with how the best or most appropriate combination of model and algorithm can be identified. This question was addressed in Chapter 6 – see also Chapter 7 and Chapter 8.

9.3.1 Achievements

In Chapter 6, it is proposed that a performance measure for region tracking algorithms is computed by regarding the algorithms as classification systems and computing their ROC (receiver operating characteristic). An experimental protocol, by which these algorithms can be compared using this measure of performance is then suggested and it is shown how it can also be used to compare the performance of the underlying statistical models. Application of this protocol to the problem of selecting the best region tracking algorithm from a set of candidates and for the selection of the most appropriate statistical colour model for two separate applications is documented in Chapter 7.

The results obtained often agree with intuitive expectations from theoretical considerations or inspection of algorithm output, but have also produced some surprises. The quantitative comparisons of both the region tracking algorithms and the underlying statistical models have greatly increased the understanding of each and allowed firm conclusions about their properties to be made. For example, that distributions of chromaticity are better modelled by distributions on the unit sphere than on fixed planes; that directional statistical models do not generalise well to distributions of colour data obtained from daylight scenes; and that more stable active contours are more susceptible to local minima.

9.3.2 Further Work

As noted in Chapter 8, the integrity of all of the results obtained here would be enhanced by the use of larger training and test sets and also by the use of a validation set. However, the process of extracting ground truth images for large sets of images is very laborious and the computer time required to obtain ROC curves over large data sets can also be a problem. Broad
distinctions have been made successfully over the data sets used here, but for more reliable comparison of some of the more similar models larger test sets are required. The measure of uncertainty in the performance measure, proposed in section 7.2.4, was also discussed in section 8.4 and it was concluded that this measure may not fully capture the uncertainty. Further investigation into the precision provided by the performance measure might be valuable. If a binormal model for ROC curves is assumed (i.e., the two underlying distributions shown in Figure 38, section 6.5.2, are Gaussian — see Appendix C), then the maximum likelihood parameter fitting procedure originally proposed by Dorfman and Alf, [DA69], can be used. This procedure can be extended to provide a measure of the goodness of fit of the binormal model to the sampled ROC points, [MKW85], which can also provide an estimate of the uncertainty in the performance measure, z(A). Unfortunately, the Newton method used to find the maximum likelihood parameter estimates is unreliable and has often been found not to converge. Alternative techniques for this iterative estimation procedure could be a valuable direction for the future.

The performance characterisation procedure is a lengthy process and cannot be used to select the most appropriate from a set of candidate statistical models on-line. As discussed in the conclusion to Chapter 3, there are other ways to obtain measures of quality for statistical models, such as the information criteria of Akaike, or Rissanen. Such measures can be obtained very quickly and could be used to select the most appropriate from a set of candidate models on-line. However, these measures are not necessarily reliable and in order to be trusted the decisions made using one of these techniques should be shown to tally well with those made using a more reliable but lengthy process such as that used in this work. A potentially useful direction for future work would be to investigate the reliability of these measures.

One further possible extension to the basic ROC theory used in this thesis, which would be useful for analysing similar systems to those used here, would be the incorporation of further classification options. In particular, a doubt class that, rather than being ignored completely as it has been here, makes some contribution to the overall measure of performance. It is not hard to envisage algorithms similar to those used here that assign a doubt category as well as positive and negative classifications and a measure of performance of such an algorithm should include the level of agreement with the doubt category in the ground truth.

9.4 Other Possible Directions

Another useful extension to the work presented here might be to extend the range of region tracking algorithms that have been compared. Of particular interest would be a comparison of the statistically based algorithms used here with more traditional edge seeking algorithms such as active contour models. These algorithms contain no tolerance parameter, k, and so an
alternative parametrisation of the ROC curve would have to be sought. However, whichever parameter is used to parametrise the curve, measures taken from it should still be comparable with those computed for the algorithms in this thesis.

As discussed in the introduction, for dependable performance of a tracking system, information should be used from every available source. The statistical approach taken in this work lends itself well to future extension of the colour-based systems discussed here by the incorporation of other sources of information. Statistical measures from underlying models of many different types of data might be combined (see [AG92] for some discussion of techniques for fusing information in this way) to generate a more complex goodness function. Characterisation of the performance of such a system may prove more complex however as there are may be many parameters that can affect the ratios of true and false positives.
References


CA79  GB Coleman, HC Andrews. "Image Segmentation by Clustering". Proceedings of
References


References


References


References


References


References


References


References


202
References

Appendix A Multi-mode SCMs

The SCMs in Chapter 3 are designed for a single cluster of colour data arising from an object of essentially one surface reflectance, under an illumination of constant chromaticity. Although for such distributions these models may be very accurate, as soon as these constraints are relaxed, they become less appropriate. When the object of interest is multicoloured, the distribution of colour data tends to separate into distinct clusters or modes. If the changes in colour over the surface of the object of interest are smooth, then the distribution may not be clustered but simply more widely spread. When the changes in colour are sudden however, and the image regions corresponding to the differently coloured regions of the object are large compared to the boundaries between these regions, the distribution becomes multimodal.

Multiple sources of illumination can also cause clustering of the distribution. As discussed in section 2.1.5, when there are multiple sources of illumination with different chromatic properties, the direction characterising the chromaticity from a single surface patch varies depending on the balance of the different illuminants incident on the patch. However, the set of directions is restricted to a polygon on the unit sphere with vertices corresponding to the patch viewed under pure illumination from each individual source. Consider first an object with a regular shaped surface for which the distribution of surface facet orientations is uni-modal. If the light sources are diffuse, the balance of illuminants over the surface varies smoothly from point to point causing the observed distribution of chromatic properties to be uni-modal. When some or all of the light sources are more directional, the distribution of reflected light can become clustered, because of shadowing or abrupt boundaries to the range of the surface over which light from a particular source is incident. In such situations some clustering of the distribution of chromaticity features can be expected.

Clustering in the RGB space can also occur because of distinct peaks in the intensity of the reflected light rather than the chromaticity. This situation might occur, for example, for an L-shaped object for which the distribution of surface facet orientations is multimodal. If such an object is viewed under multiple illuminations, the chromaticity distribution may also be multimodal whether the light sources are diffuse or directional. There are other effects such as specularity and interreflection that may also cause clustering of the distribution to occur.

When distinct clusters are formed, a single mode model cannot cover all the clusters of the real distribution without also covering regions in between the clusters. Data in these regions, however, may only very rarely arise in the region of interest rendering the model inaccurate. When clustering does occur, more complex models that deal with distinct clusters separately are more appropriate and such models are the subject of this appendix.
Figure 70 shows an image from a sequence in which the distribution of data corresponding to the region of interest may not be well modelled by a distribution with a single mode. There are several multicoloured objects in the image but one is selected to illustrate the ideas in this appendix and has a seed region outline superimposed on it. The object of interest consists of several differently coloured regions and, in a similar way to the image on the left in Figure 3, in Chapter 3, is illuminated by a single light source in an otherwise darkened room.

Figure 70 A multicoloured object under a single illumination (left) on a cluttered background.

Figure 71 Occurrence histograms of colour data showing the sample (left) and parent (right) distributions from the object of interest in Figure 70.

Figure 71 shows the sample and parent distributions from the region of interest in Figure 70. Single model trivariate Gaussian and directional chromaticity uniform intensity (in fact BCUI) models are depicted in Figure 72. The chromaticity distribution has high variance
Appendix A Multi-mode SCMs

causing the directional model to spread very widely about its mean. The Gaussian model appears to be a closer approximation to the parent distribution but some regions of the colour space are given unnecessarily high likelihood.

![Figure 72 Single mode Gaussian (left) and BCUI (right) models constructed from the seed region shown in Figure 70.](image)

When the distribution of colour data does form distinct clusters, the single mode SCMs of Chapter 3 can be used to model each separate cluster in the data. These models can then be combined to form a multimode model of the region as a whole. For this multimode model to be accurate, the sample distribution from which the model is constructed must form similar clusters to the parent distribution. The only problem that remains is that of dividing the sample distribution into distinct clusters.

A.1 Clustering

Given a sample set of data from a seed region in an image, the sample must first be separated into clusters. There are several levels of difficulty of the general clustering problem, [DH73]. The problem in its most general form is to divide a given distribution of data into separate clusters where the number of clusters is unknown and only minimal information about the shape of the clusters, such as that they form connected regions in the measurement space, is available. Often, assumptions can be made about the shape of the clusters, i.e., each cluster is known to follow some parametric distribution, for example a Gaussian distribution, and this simplifies the problem to some extent. Also, knowledge about the number of clusters contained in the data may also be available and this too simplifies the cluster extraction procedure. The simplest form of the problem occurs when both the number of clusters and their shape are known. Jain, [Jai86], provides a brief review of clustering techniques aimed at pattern recognition and image processing applications. Kaufman and Rousseuw’s book, [KR90], is a practical guide to many common clustering techniques and provides sample code for most. Jain and Dubes, [JD88]
Appendix A Multi-mode SCMs

provide a more comprehensive review and further general discussion and review can be found in [DH73, DK82, Rip96].

Many segmentation techniques are based on clustering procedures, the underlying philosophy being that data from the same image region occupy similar regions in measurement space. The clustering algorithm is used to identify separate clusters in the measurement space and the resultant separation of the data is projected back to separate the image itself. Construction of multimode statistical models is a similar problem, although separate image regions no longer have to be explicitly identified. Furthermore, when constructing a statistical model, the amount of data provided is typically much smaller, as it is only that contained in a seed region rather than an entire image.

The k-means algorithm, [DH73], is a common technique for solving the simplest form of the clustering problem. To apply the k-means algorithm, both the number of clusters in the data, \( k \), and the shape of those clusters must be specified. Cluster shape is given by a metric on the measurement space usually derived from a probability density function; for example, if the clusters are Gaussian, the Mahalanobis metric is used. Crisman, [Cri92], and Thorpe, et al, [THK88], used the k-means algorithm to construct statistical colour models. Two separate models are created - one for regions of interest and one for the background. The k-means algorithm is used to cluster sample data from each of the two regions separately into 4 Gaussian clusters with independent parameters (the covariance can vary between clusters). The algorithm provides the parameters of the single mode models of each cluster, which in turn define the multimode model, as well as an assignment of each sample to a particular cluster.

In general the number of distinct regions in an image, or seed region, is unknown and so the number of clusters required cannot be pre-specified and must be extracted from the data. One way to address this problem is to perform the clustering with various values for the pre-specified number of clusters and use some other criterion to choose which clustering is best. Coleman and Andrews, [CA79], use a k-means clustering algorithm for image segmentation where \( k \) is varied over a number of values. Clusters are assumed to be Gaussian. A good clustering is implicitly defined as one that minimises the variance of data within individual clusters, while maximising the separation of the clusters or between cluster variance. Several criteria, all based on numerical measurements of the within cluster and between cluster variances, are proposed to identify the value of \( k \) that produces the best clustering of the data. Coleman and Andrews algorithm also selects, from a large number of image features, including intensity, colour and texture measurement, which ones should be used to determine the image segmentation. Results on several images are not conclusive, but appear to show that, on at least one image, when an appropriate set of image features has been selected, their criterion selects the right number of clusters.
Appendix A Multi-mode SCMs

When a model of the shape of each cluster found in the data is assumed and there are parameters in the clustering procedure controlling the number of clusters extracted, information criteria such as the AIC, discussed in section 3.8, can be used to select optimum clustering levels. Zhang and Modestino, [ZM90], use the AIC in conjunction with the k-means clustering procedure, in exactly this way, in their model based image segmentation technique. At a number of different settings for k, the clustering is performed assuming Gaussian (intensity only) models for each cluster. The clustering yields a statistical model of the data found in the image, from which the AIC can be calculated. The value of k for which the AIC is minimised indicates the true number of clusters found in the data and is used to perform a segmentation of the image. Over some artificially generated clustered data, the AIC is shown to select the correct number of distinct clusters. Further experiments show correct identification of the number of regions in artificially generated textured images. The segmentation algorithm based on this clustering is applied to two aerial photographs and the output is shown. The number of clusters identified and the resultant segmentations seem reasonable for both images.

Haralick and Kelly, [HK69], proposed an alternative clustering procedure to be used for the segmentation of multiband (though not the familiar colour bands) aerial photographs. Although the number of clusters must still be specified in their scheme, no assumption about the shape of the clusters is made. They describe how the procedure can be applied in either the spatial domain of the image, when in fact it becomes a region growing algorithm, see Chapter 4, or in the measurement space. Results are shown on an aerial image and major regions are extracted effectively when internal parameters, controlling the number of regions, are appropriately set.

Khotanzad and Bouarfa, [KB90], use the graph-theoretic clustering procedure suggested by Koontz, et al, [KNF76]. This procedure assumes no knowledge of the number of clusters in the data or the shape of the clusters. However, the resolution of the feature space can be varied, greatly effecting the number of extracted clusters, and must be specified by hand. In, [KB90], criteria similar to those of Coleman and Andrews are proposed for automatically determining the optimum feature space resolution at which to perform the clustering, making the entire procedure fully automatic. A segmentation algorithm is based on this clustering procedure where clusters are validated in the spatial domain after having been extracted in the measurement space. A simple connectivity measurement of the image region corresponding to each cluster is made and clusters whose connectivity falls below a certain threshold are rejected and merged with neighbouring clusters. Results on four artificial textured images show correct identification of the number of clusters in each, with a few small noisy clusters being eliminated by the cluster validation phase. Results are also shown for a single grey-scale image and a single colour image, which appear to be reasonable segmentations.
Appendix A Multi-mode SCMs

Often different regions of an image can have very similar properties but can be clearly distinguished by being well separated spatially. Segmentation algorithms based on clustering properties of pixel data alone cannot separate such regions. Matas et al, [MMK95, Mat96], developed a segmentation algorithm similar to that of Khotanzad and Bouarfa. The same clustering procedure was used but a more sophisticated, spatial, cluster validation and refinement procedure is proposed. Connected components of the regions corresponding to measurement space clusters are extracted and their statistics examined separately. If two separate connected components are found to have significantly different statistics, the cluster is split. Efficacy of the approach is shown on some synthetic data and segmentations of real images using the clustering procedure both with and without the spatial refinement are shown. As might be expected, segmentations obtained using the spatial refinement scheme are comparatively less noisy in appearance than those obtained using the clustering procedure alone.

As in the general segmentation problem, when creating SCMs, ideally, the number of clusters in the sample data should be extracted automatically. This allows complex models to be generated for highly complex regions of interest, while for simple regions consisting of one or two different colours, simpler models are constructed avoiding over fitting. Given the theory and set of candidate SCMs proposed in Chapter 3, in certain circumstances it would be reasonable to assume that the shape of each cluster is known. A k-means algorithm could then be used together with either a cluster separation measurement, as in [CA79], or an information criterion, as in [ZM90], to select the number of clusters. For a given application, the most suitable model for single mode regions of interest under the lighting conditions encountered in that application could be found by experimentation, e.g., in the way described in Chapter 6 and Chapter 7, and then the same model used for each cluster of multimode distributions. However, it is considered preferable here, for reasons of generality and consideration of future work where different models might be selected for different clusters, not to impose any a priori constraints on the shape of the clusters. Several clustering methods are available in which no assumptions concerning cluster shape are made, for example, [NG77, KNF76, Wha83, Kit76], but it is not clear which, if any, is likely to produce the best results. No comparison of these clustering procedures has been performed, but because of the reasonable results reported in both [KB90] and [MMK95, Mat96], the clustering procedure of [KNF76] with the automatic parameter selection techniques of [KB90], has been adopted here. The spatial cluster validation and refinement methods suggested in [KB90] and [Mat96] are not used, as spurious clusters are less of a problem when building statistical models rather than producing a full segmentation. However, a minimum threshold on cluster size is enforced, below which the cluster and the data contained in it is ignored. Details of the algorithm and implementation are given in Appendix B, section B.3.
Appendix A Multi-mode SCMs

A.2 Statistical Modelling

Once clustering has been achieved, a statistical model must be created. The parameter estimation phase is just like the parameter estimation of the single mode models described in the previous chapter. For each separate cluster found, the parameters of a model are found by maximum likelihood estimation using the data assigned to that cluster.

There are two possible ways to derive goodness and likelihood measures, from SCMs constructed in this manner. Both methods involve finding a measure of likelihood from the model of each cluster that the pixel under investigation is part of that cluster. Suppose \( n \) clusters, \( \omega_1, \ldots, \omega_n \), are found in a sample distribution of data and each is modelled by a distribution with probability density function, \( f_i(x) \), where, \( i=1\ldots n \). Suppose further that the total number of data points in the sample is \( M \), and that each cluster contains \( m_i \) of those data, so that \( M = \sum_{i=1}^{n} m_i \). The a priori probability that a new datum, \( x \), is part of the \( j \)-th cluster, given that it is part of the region of interest, \( RI \), is then,

\[
P(x \in \omega_j | RI) = \frac{m_j}{M}.
\]  

Using the standard mixture model for multimode distributions, [DH73], the following probability distribution for \( x \), given that the pixel is part of the region of interest is obtained:

\[
p(x | RI) = \sum_{i=1}^{n} \frac{m_i}{M} f_i(x).
\]

Unlike the single mode models of Chapter 3, deriving a distance function from this probability density function is not straightforward. If the logarithm of the density function is taken, and constant terms subtracted, in the usual way, a function is obtained that never becomes zero. Thus, alternative definitions for the goodness and likelihood, c.f. (17) and (18), must be devised. Two approaches are investigated here; both involve finding a measure of likelihood from each cluster that a piece of data is part of that cluster. A likelihood, \( L_i \), that a new datum belongs to a particular cluster may be found by considering that cluster as a single mode SCM and calculating the likelihood in the usual way, thus,

\[
L_i(x | x \in \omega_i) = \exp \left\{ -\frac{1}{k} d_i(x, \bar{x}) \right\}.
\]

where \( d_i \) is a metric derived from \( f_i \), and the same \( k \) is used for each cluster.

A simple way to find an overall likelihood measure, following Crisman, [Cri92], and Thorpe, et al, [THK88], is to assume that the datum comes from the cluster for which \( m_i / M \log(L_i(x | \omega_i)) \) is greatest and then to define the overall likelihood, \( L \), to be
Appendix A Multi-mode SCMs

\[
L(x) = \exp\left(\frac{1}{M} \log(L_i(x|\omega_i))\right).
\]  

(78)

The goodness measure, of (17), may then be defined in the usual way, see (17) and (18), i.e.,

\[
G(x) = 1 - \log L(x).
\]  

(79)

Although this method is appealingly straightforward, it is a simplification of the mixture model in that only one cluster contributes to the overall likelihood. The second method is more intuitively appealing as it adheres more closely to the standard mixture model. The likelihoods associated with each cluster are defined as in (77), but now each cluster contributes to the overall likelihood, \(L\), which is defined as follows,

\[
L(x) = \exp\left(\sum_{i=1}^{n} \frac{P(\omega_i | R) f_i(x) \log L_i(x)}{\sum_{i=1}^{n} P(\omega_i | R) f_i(x)}\right) = \exp\left(\sum_{i=1}^{n} \frac{m_i f_i(x) \log L_i(x)}{\sum_{i=1}^{n} m_i f_i(x)}\right).
\]  

(80)

This provides an intuitive expression for the likelihood, \(L\), as its logarithm is the average of the log likelihoods returned by each single mode cluster model weighted by the probability that the data is from that cluster. From this definition, a similarly intuitive definition for the goodness, \(G\), is obtained:

\[
G(x) = 1 - \log L(x) = 1 - \frac{\sum_{i=1}^{n} m_i f_i(x) \log L_i(x)}{\sum_{i=1}^{n} m_i f_i(x)} = \frac{\sum_{i=1}^{n} m_i f_i(x) (1 - \log L_i(x))}{\sum_{i=1}^{n} m_i f_i(x)}
\]  

(81)

i.e.,

\[
G(x) = \frac{\sum_{i=1}^{n} m_i f_i(x) G_i(x)}{\sum_{i=1}^{n} m_i f_i(x)}.
\]  

(82)

where \(G_i\) is the goodness obtained from the single mode model of cluster \(i\), exactly as in (17). In this approach, the value of the probability density function \(f_i(x)\) has to be included in each contribution to the sum in the expression for the likelihood or goodness so that the overall value is not dominated by contributions from large clusters from which a particular piece of data is very distant. Although the resulting expression has similar form to the mixture model probability density function, it is not derived directly from it.

It is not clear which of these two approaches provides the best model. The second method assigns higher probabilities to pixels with values that lie in between two distinct clusters. If there are boundaries in the region of interest between areas consisting of pixels from
two distinct clusters, the pixels lying on that boundary tend to take values that lie somewhere between the two clusters, some amount of each type of incident light reaching that point of the sensor array. If such regions are sufficiently represented in the seed region however, the first approach may also cope well with such data.

All the multimode models investigated here used the same generic single mode model for each cluster. The parameters of the model, however, are allowed to vary arbitrarily from cluster to cluster.

A.3 Preliminary Results

Figure 73 shows fixed likelihood contours from multimode SCMs, Gaussian and BCUI, constructed using the procedure described above for the object of interest in the image of Figure 70. Nine clusters were extracted from the sample distribution in the seed region in the image. Little difference can be observed between pictures of the full multimode SCMs or the chromaticity models obtained using different definitions for the likelihood, but the ones shown in fact use the second. The sample and parent chromaticity distributions are depicted in the unit plane in Figure 74 and single mode and multi-mode chromaticity models constructed from that sample are shown in Figure 75.

ROC curves for single mode and multi-mode Gaussian models, generated using the thresholded likelihood image algorithm, described in section 5.1, with no morphological post-processing step, are shown in Figure 76. The two types of multi-mode model are numbered in the order they were presented in the previous section, thus, “Multi-mode 1” denotes the simpler likelihood definition which uses just one cluster, and “Multi-mode 2” denotes the definition in which contributions are summed over all the clusters. Similar figures for Bingham chromaticity, uniform intensity and Bingham chromaticity, Gaussian intensity models are shown in Figure 77 and Figure 78, respectively. Overall measures of performance for each model are given in Table 16.

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Model Type</th>
<th>Z(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian</td>
<td>Single Mode</td>
<td>0.813247</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Multi-mode 1</td>
<td>0.809594</td>
</tr>
<tr>
<td>BCUI</td>
<td>Single Mode</td>
<td>0.808967</td>
</tr>
<tr>
<td>BCUI</td>
<td>Multi-mode 1</td>
<td>0.808335</td>
</tr>
<tr>
<td>Gaussian</td>
<td>Multi-mode 2</td>
<td>0.804789</td>
</tr>
<tr>
<td>BCUI</td>
<td>Multi-mode 2</td>
<td>0.798461</td>
</tr>
<tr>
<td>BCUI</td>
<td>Multi-mode 2</td>
<td>0.643665</td>
</tr>
<tr>
<td>BCUI</td>
<td>Multi-mode 1</td>
<td>0.632319</td>
</tr>
<tr>
<td>BCUI</td>
<td>Single Mode</td>
<td>0.571057</td>
</tr>
</tbody>
</table>

Table 16 Values of Z(A) found for each model.
Appendix A Multi-mode SCMs

Figure 73 Multimode statistical colour models for the region of interest in the image shown in Figure 70.

Figure 74 Sample (left) and parent (right) chromaticity distributions from the region of interest in Figure 70.

Figure 75 Single mode (left) and multi-mode (right) Bingham chromaticity models for the region of interest in Figure 70.
Appendix A Multi-mode SCMs

**Gaussian models**

Figure 76 ROC curves for single mode and multi-mode Gaussian models.

**BCUI models**

Figure 77 ROC curves for single mode and multi-mode BCUI models.
Appendix A Multi-mode SCMs

Figure 78 ROC curves for single mode and multi-mode BCGI models.

A.4 Discussion

For both the Gaussian and BCGI based models, the simpler definition of the likelihood appears to work slightly better for the multi-mode models. However, in both cases, single mode models produce better overall classification performance. Both of these sets of curves follow a similar pattern. For low operating points, the multi-mode models perform better, having lower false positive ratios, as might be expected because the single mode models cover regions of the measurement space not occupied by the parent distribution. However, for moderate operating points, the single mode models provide significantly better performance and it must be concluded that the multi-mode models are over specific to the sample data and do not generalise as well to unseen data as the single mode models. This conclusion is supported by some extent by the large number of clusters that are extracted. A related reason for the lower performance of the multi-mode models is that the seed region is small and does not accurately represent the parent distribution. From Figure 74 it can be seen that the large green chromaticity cluster in the parent distribution is poorly represented in the sample. Single mode models that are spread wide about their mean are likely to encounter more distant regions of the measurement space, such as this green part, at lower operating points than the multi-mode models, which fit the sample distribution more tightly. In general, at corresponding operating points, the true positive ratio for multi-mode models using the simpler likelihood definition is slightly higher than for the
other multi-mode models and the false positive ratios are approximately equal. The reason for this may be that the simpler model assigns higher likelihood to pixels with values close to the mean of a distinct cluster than the other model, which assigns slightly lower likelihood in these regions and slightly higher likelihood to regions of the measurement space between clusters.

The performance of the uniform intensity models is very poor, but the order of increasing performance is the complete reverse of the other models. This is probably accounted for by the large percentage of the measurement space covered by these models, particularly the single mode model, whose overall classification performance is little better than random chance. The simpler multi-mode models performs better at lower operating points, as might be expected as the model is tighter around distinct clusters so reducing the false positive ratio. For moderate operating points it is slightly worse however, perhaps extending to the poorly represented green part of the parent distribution less quickly.

The work on multimodal models presented in this appendix is far from complete and there is much further work to be done. A comparison of some other clustering techniques would certainly be useful. An interesting possibility is the use of information criteria, such as the AIC, in conjunction with the clustering procedure used here to select the histogram resolution at which the clustering is performed. Any of these simple measures could be used in this way and may produce more reliable results than the current method, which is based on cluster density and isolation and does not consider the statistical model at all. Moreover, using information criteria, or other methods, it may be possible to select individual single mode models for separate clusters. The multimode models discussed here use the same single mode distribution to model each extracted cluster but an information criterion could be used to select the most appropriate from a set of candidates for each separate cluster. The resulting multimode model could consist of any number of different single mode distributions. As noted in section 3.8, good correspondence between model selection by full performance characterisation and use of an information criterion must be shown before the information criterion can be fully trusted. One disadvantage of such an approach is that the statistical model has to be explicitly calculated for each candidate clustering, which can be computationally expensive. In the method used here, the statistical model is only constructed once the optimal clustering has been identified.
Appendix B Algorithms

Appendix B Algorithms
This appendix contains outlines of various algorithms used in the work described in the preceding chapters. Details of these algorithms are included mainly for the purposes of reproduction of the work contained in the body of this thesis and are not intended for the more casual reader.

B.1 Maximum Likelihood Estimation of Intensity Maximum
This section outlines the procedure for finding a maximum likelihood estimate of the \( \text{maxI} \) parameter in the planar surface intensity, PSI, model – see equation (45), in section 3.4.3.

It is required to find a setting for \( \text{maxI} \) such that the product of the probabilities assigned to each member of the sample distribution is maximised. Equivalently, the sum of the logarithms of these probabilities can be maximised, as is common in maximum likelihood estimation. The idea is to compute \( \text{maxI} \) by assuming that there is a single maximum likelihood point. Three estimates are thus maintained so that the vicinity of a maximum point can be detected and the estimates iteratively moved closer to that point. If a minimum point is detected, which indicates that there are multiple maxima, the maximum point with the higher value of \( \text{maxI} \) is selected.

- Set \( \text{maxsamp} \) to the maximum intensity of all the data in the sample distribution.
- Three initial values of \( \text{maxI} \) are defined: \( \text{minmaxI} \), \( \text{midmaxI} \) and \( \text{maxmaxI} \).
- Set these values initially to \( \text{maxsamp} \), \( 2 \times \text{maxsamp} \), \( 4 \times \text{maxsamp} \), respectively.
- For each setting, the intensity model is constructed in the usual way – see 3.4.3.
- The log likelihood is summed over the sample distribution for each of the three models giving values \( \text{liksummin} \), \( \text{liksummid} \), \( \text{liksummax} \), corresponding to each of the three \( \text{maxI} \) settings respectively.
- If \( \text{liksummin} \) is less than \( \text{liksummid} \),
  - If \( \text{liksummid} \) is less than \( \text{liksummax} \),
    - Set \( \text{minmaxI} \) to \( \text{midmaxI} \), \( \text{liksummin} \) to \( \text{liksummid} \).
    - Set \( \text{midmaxI} \) to \( \text{maxmaxI} \), \( \text{liksummid} \) to \( \text{liksummax} \).
    - Add the difference between \( \text{maxmaxI} \) and the old \( \text{midmaxI} \) to \( \text{liksummax} \).
    - For the new \( \text{maxmaxI} \) reconstruct the model and recalculate \( \text{liksummax} \).
  - Otherwise
    - Find the \( \text{maxI} \) setting, either \( \text{minmaxI} \) or \( \text{maxmaxI} \), which gives the lowest log likelihood sum, halve its distance to \( \text{midmaxI} \). Reconstruct the corresponding model and recalculate the log likelihood sum.
- Otherwise
Appendix B Algorithms

- If `liksummid` is less than `liksummax`,
  - Set `minmaxl` to `midmaxl`, `liksummin` to `liksummid`.
  - Set `midmaxl` to `maxmaxl`, `liksummid` to `liksummax`.
  - Add the difference between `maxmaxl` and the old `midmaxl` to `maxmaxl`.
  - For the new `maxmaxl` reconstruct the model and recalculate `liksummax`.
- Otherwise
  - Set `maxmaxl` to `midmaxl`, `liksummax` to `liksummid`.
  - Set `midmaxl` to `minmaxl`, `liksummid` to `liksummin`.
  - Subtract the difference between `minmaxl` and the old `midmaxl` from `minmaxl`.
  - If `minmaxl` is less than `maxsamp`
    - Set `minmaxl` to `maxsamp`.
  - Otherwise
    - Set `midmaxl` to `(maxsamp + maxmaxl)/2`.
    - For the new `minmaxl` reconstruct the model and recalculate `liksummin`.
- Repeat the previous step until the difference between `minmaxl` and `maxmaxl` is less than some fixed threshold; in this work 1.
- Set `maxl` to `midmaxl`

Even though `maxl` must be at least as high as the intensity of any data in the sample, it is possible that the value for `maxl` could be less than the maximum possible intensity attainable, \(255^3\) in the standard RGB colour cube. This allows the possibility that a piece of data might be encountered whose intensity is greater than `maxl`. Such an event causes problems in equation (45) as the inverse cosine of a value greater than one is required. In practice when the intensity of some piece of data is found to be greater than the value of `maxl`, the value is replaced by `maxl` and the likelihood calculated accordingly.

**B.2 Statistical Colour Models**

This section describes some general points concerning the implementation of statistical colour models such as those in described in Chapter 3 or Appendix A.

**B.2.1 Black Pixels**

For pixels whose value is zero in each of the three colour bands, i.e., whose value lies at the origin of the RGB cube, the chromaticity, defined as the direction of the ray from the origin to the RGB value, is not defined. When confronted with such a pixel however, some likelihood value needs to be assigned by the SCM, so that an algorithm based on that model does not break down. The choice of likelihood assigned is somewhat arbitrary, but in this thesis the distance of
Appendix B Algorithms

A black pixel from the mean is defined to be equal to the tolerance parameter, k, see equation (17). This definition ensures that the goodness, $G$, of a black pixel is zero for any setting of $k$. This scheme is used in all the directional chromaticity models described in section 3.3 and multimode versions of those models described in Appendix A.

B.2.2 Look-up Tables

In a typical region tracking task, the set of pixel values encountered by the SCM at the heart of the region tracking algorithm is quite restricted. Furthermore, once a pixel with one set of colour values has been encountered, it is likely that similar pixels with the same values will be encountered in the future. The computational performance of the statistically based region tracking algorithms described in Chapter 5, might therefore be enhanced by use of a look-up table. Once the likelihood or goodness for a particular RGB vector has been calculated, it is stored in a look-up table so that it does not have to be recalculated if similar data appear in the future. The look-up table is stored as an octree, in which a new node is created whenever an unseen RGB vector is encountered. Although the use of such a data structure for the look-up table adds some complexity over the use of say a simple 3D array, as the actual set of RGB vectors encountered is relatively small compared to the whole RGB cube, there are significant savings in terms of storage space.

Significant improvements in computational requirements are achieved using this scheme, particularly for the more computationally intensive models like the Bingham based and multimode models. In fact the values that are stored in the look-up table in the statistical colour models used here are the distance measures, see section 3.1, used in the definitions of likelihood and goodness, which is independent of $k$ – the tolerance parameter. Thus when an algorithm is applied to a task using several values of $k$ consecutively, for example when computing the receiver operating characteristic of the algorithm, the look-up table can be retained for separate runs through the test image sequence.

B.2.3 Computation Times

In this section the computational load imparted by the different statistical models used throughout this thesis are compared. The processor time required for construction of likelihood images from two consecutive frames of a sequence of images, in fact the first two frames from the sequence of images containing the track in Figure 3, using a variety of different SCMs are given in Table 17. The tolerance parameter is non-zero for each model and times are quoted for each model with and without use of the look-up table described in the previous section on each image. The two images are analysed consecutively so that the look-up table built up on the first image is used while the second is examined. Exactly the same number of accesses to the SCM is
made for each separate model. The processor time for construction of each SCM from the seed region, which contains approximately 20000 pixels, is also quoted in Table 17.

For all of the models, apart from the model incorporating the planar surface intensity model, the construction times are similar. The complexity of the maximum likelihood estimation phase for each of these models is of the order of the number of sample pixels, but there is a slight overhead with the Bingham based models. The parameter estimation phase for the planar intensity model is iterative and involves repeated construction of the intensity model, see section B.1, which increases the required processor time. When likelihoods are computed for the same set of pixels, the Gaussian based SCMs all have similar computational performance. The Bingham models appear to require approximately four times as much processor time. The use of the look-up table has only a slight effect on the computation times required by the Gaussian based models, although by the end of the sequence, once the table is well established, computation time is reduced to around 12s per frame. The effect is more noticeable in the Bingham models, for which computation time is approximately halved on the first image and is further reduced for the second. The effect of the intensity models on likelihood computations appears to be small, even for the planar intensity model.

<table>
<thead>
<tr>
<th>SCM</th>
<th>Construction time</th>
<th>With look-up table</th>
<th>Without look-up table</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Frame 1</td>
<td>Frame 2</td>
</tr>
<tr>
<td>3DG</td>
<td>2s</td>
<td>18s</td>
<td>17s</td>
</tr>
<tr>
<td>BCUI</td>
<td>3s</td>
<td>42s</td>
<td>31s</td>
</tr>
<tr>
<td>BCGI</td>
<td>3s</td>
<td>44s</td>
<td>33s</td>
</tr>
<tr>
<td>BCPSI</td>
<td>20s</td>
<td>45s</td>
<td>33s</td>
</tr>
<tr>
<td>GNCCUI</td>
<td>2s</td>
<td>20s</td>
<td>18s</td>
</tr>
<tr>
<td>GOPCUI</td>
<td>2s</td>
<td>20s</td>
<td>18s</td>
</tr>
<tr>
<td>ZYCUI</td>
<td>2s</td>
<td>20s</td>
<td>18s</td>
</tr>
<tr>
<td>ANCUI</td>
<td>2s</td>
<td>21s</td>
<td>18s</td>
</tr>
<tr>
<td>HCUI</td>
<td>2s</td>
<td>20s</td>
<td>18s</td>
</tr>
</tbody>
</table>

Table 17 Processor times for likelihood image construction using various SCMs. All times are quoted in seconds.

B.3 Clustering Procedure

This appendix presents details of the clustering algorithm used in the construction of the multimodal statistical colour models discussed in Appendix A. The clustering procedure used is the histogram based method of Koontz, et al, [KNF76]. The algorithm proceeds as follows:

- Generate a histogram of the sample data.
Appendix B Algorithms

• For each non-zero histogram bin, find the bin with the largest count in the direct neighbourhood of that bin. The direct neighbourhood is the set of bins that correspond to adding or subtracting one to any number of the components in the data and includes the bin itself. Thus if the data is of dimension n, the neighbourhood contains 3^n bins. Create a link from the bin under investigation to the one with the largest count.

• Identify a set of peak bins by finding bins that are linked to themselves and so are local maxima.

• If two peaks are found to be neighbours, which can occur if they have the same count, one of the two is chosen arbitrarily and the other linked to it.

• Each peak defines a separate cluster in the histogram.

• By following the links created for every non-peak bin, eventually one of the peaks will be found. The set of data linked to a particular peak comprises the cluster associated with that peak.

The basic algorithm outlined above is conceptually very simple. There are however, some implementational details and general enhancements that should be mentioned.

B.3.1 Histogram Storage

When constructing the histogram, it may not be feasible do so by allocating a bin for every possible position in the measurement space particularly if that space has high dimension. Even for the RGB colour cube, which has fairly low dimension, the memory requirement is equivalent to roughly 16 million integers. The histogram is usually only sparsely populated and so bins are only created when their count becomes non-zero. Accessing bins, however, now becomes more troublesome. If the bins are stored, for example, in an unordered linked list, access times are of the order of the size of the sample distribution, which is impractical for reasonably large sample distributions. One approach might be to impose an ordering on the list of non-zero bins to reduce access times to the order of the log of the size of the sample. A simpler approach has been adopted here, however, which is the use of a hash table. The hash table used has 1000 elements each of which is a pointer to a linked list along which a linear search is performed for access to a particular bin. The hashing function used takes the last digit of each component integer measurement and concatenates them to give a number between 0 and 999. This function has proved to provide extremely even coverage of the table ensuring greatly improved access times.

B.3.2 Histogram Resolution

The resolution of the histogram into which the sample is put can have a profound effect on the number of clusters extracted by the algorithm outlined above. If the resolution is too high, many spurious clusters are detected containing a very small number of samples. A model constructed from such a clustering would almost certainly be over-fitted to the sample and generalise badly.
Appendix B Algorithms

to unseen data. For the sample distributions encountered in this work, using the full resolution of the RGB colour cube produces far too many clusters and better results are obtained by lowering the resolution. However, if the resolution is too low, distinct clusters can be merged into one. Khotanzad and Bouarfa, [KB90], proposed two criteria for automatic selecting of the resolution that produces the best clustering.

B.3.2.1 The "S" Method

The first criterion, called the S method, is recommended for low dimensional (1-3D) data. It is designed to maximise both the density and isolation of each individual cluster. The clustering procedure outlined above is performed at each possible histogram resolution and a measure is calculated for each. The density of a cluster is defined to be the number of samples in that cluster divided by the number of histogram bins they occupy. A measure of compactness of a cluster is then found by finding its nearest neighbouring cluster, in terms of Euclidean distance between peak histogram bins, and taking the absolute difference in their densities divided by the distance between the peaks. To obtain a measure of quality for the clustering as a whole, this measure is summed over each extracted cluster. The clustering that maximises this measure is selected as the best.

B.3.2.2 The "M" Method

The M method is recommended in [KB90] for higher dimensional (>3) data. The criterion is simply defined to be the ratio of the number of non-empty cells with a count greater than one to the number with a count equal to one. The best clustering is taken to be the one performed in the histogram with the resolution that maximises this ratio.

Unlike the S method, the M method does not actually require that the clustering be performed and so should make the whole procedure quicker. For clustering in the RGB colour cube however, the S method was found to give intuitively better results and so has been adopted throughout. The resolution of each axis is varied over the range {1, 2, 4, 8, 16, 32, 64, 128, 256}.

B.3.3 Pruning

In practice, even at the optimum histogram resolution, spurious clusters containing only very few outlying samples often arise. Because of the way the statistical model is constructed (see Appendix A) such clusters are unlikely to affect significantly the values returned by the model and their only effect is to increase overall computation times. A threshold is set on the minimum allowable size of clusters below which a cluster is deleted and the data assigned to that cluster are assumed to be outliers and so do not contribute to the construction of the statistical model. Here that threshold has been set to one thousandth of the size of the sample set. It should be
Appendix B Algorithms

noted however, that for larger sets of sample data, for example a whole image, such a fraction could represent a significant cluster.

**B.4 Region Growing**

In this section the region growing algorithm discussed in section 5.2 is detailed. A seed consisting of a set of pixels in the region of interest is provided in the first image of the sequence.

- Construct a SCM from the data found in the seed region
- The seed initially defines the object region and all other pixels are classified initially as background.
- A list of boundary pixels, defined as object pixels with one or more background pixels in their 8-way neighbourhood, is extracted from the initial classification.
- While this boundary list is not empty
  - Take the pixel at the top of the boundary list and remove it from the list
  - If the point is currently classified object, find the goodness of the pixel from the SCM
    - If the goodness is negative, set the pixel classification to background and add any pixel in the 8-way neighbourhood that is classified as object to the top of the boundary list.
    - Otherwise test all the background pixels in the 8-way neighbourhood and if they have positive goodness, classify them as object and add them to the top of the boundary list.

**B.5 Computing Goodness Integrals**

This section details the procedure for computing the line and surface integrals of a bilinearly interpolated goodness function. The surface integral of this function was used to compute estimates of the image energy change caused by the move of a single element in the SIG (Surface Integral Greedy) ARM, discussed in section 5.3.6.1.3. The line integral has not been explicitly used in the work presented in the body of this thesis, but was mentioned in section 5.3.5.2 with reference to some preliminary work on ARMs published in [AB97a]. Details of this calculation were promised in that work and so are included here for completeness.

Given a discrete 2D grid of values, such as an image, I(m,n), a continuous 2D function can be constructed over that grid by bilinear interpolation. Similarly, given a function of the image, such as the goodness, G, of equation (17), in section 3.1, which is also discretely defined, a continuous goodness function can be constructed in the same way. If GI(x,y) is used
Appendix B Algorithms

to abbreviate G(I(x,y), where x and y are real valued image co-ordinates, then the bilinearly
interpolated goodness function is given by

\begin{equation}
G(I(x, y) = (1 - t)(1 - u)GI(m, n) + t(1 - u)GI(m + 1, n) + (1 - t)uGI(m, n + 1) + tuGI(m + 1, n + 1)
\end{equation}

where m=int(x), n=int(y), t=x-m and u=y-n, [PTV92]. Integer image co-ordinates are taken to
lie at the corners of pixels only and (0,0) is taken to be the top left of the image. The function is
thus defined in a piecewise manner and is only consistent over regions completely contained
within single pixels.

**B.5.1 Line Integrals**

In this section and the next an expression for the line integral of a bilinearly interpolated surface
between points on a 2D grid is derived. In [AB97a], this integral is used as an alternative to the
discrete Bresenham line drawing approach in order to determine an average goodness value
along lines joining consecutive ARM elements.

Consider the bilinearly interpolated goodness function over a single pixel. The goodness
at a point (x,y) within that region is given by the expression in equation (83). Suppose that the
integral of the goodness function along the line joining the two points, (x₁, y₁) and (x₂, y₂),
within this region is required. The line can be expressed in parametric form as

\begin{equation}
x(s) = s, y(s) = Ms + L.
\end{equation}

where \( M = \frac{y_2 - y_1}{x_2 - x_1}, \quad L = \frac{x_2y_1 - x_1y_2}{x_2 - x_1} \) and \( s \in (x_1, x_2) \).

Substitution into equation (83) yields

\begin{equation}
G(s) = GI(x(s), y(s)) = GI(s) = \sum \left[ M^2 - (M(m + l) + n) s + (m + l)(n + 1 - L) \right]
\end{equation}

where

\begin{align*}
G_{mn}(s) &= [M^2 - (M(m + l) + n + 1 - L)s + (m + l)(n + 1 - L)]
G_{m+l,n}(s) &= [-M^2 + (Mm + n + 1 - L)s - n(n + 1 - L)]
G_{m,n+1}(s) &= [-M^2 + (M(m + l) + n - L)s - (m + l)(n - L)]
G_{m+l,n+1}(s) &= [M^2 - (Mm + n - L)s + n(n - L)]
\end{align*}

The variable of integration is changed to \( s \) by multiplication with the unit arc length,
and thus, the integral of the goodness along the line is given by

\begin{equation}
\int_{s_1}^{x_2} G(s) ds.
\end{equation}

Then, from (85),
Appendix B Algorithms

\[
\int_{x_1}^{x_2} G_I(s)ds = \int_{x_1}^{x_2} G_I(m, n) + G_I(m + 1, n) + G_I(m, n + 1) + G_I(m + 1, n + 1)
\]

All the integrals in (87) can be readily evaluated. If

\[
P = y_2 - y_1 \\
Q = x_2 y_1 - x_1 y_2 \\
R = x_2 - x_1 \\
S = \frac{1}{2}(x_1 + x_2) \\
T = \frac{1}{3}P(x_1^2 + x_1 x_2 + x_2^2)
\]

then,

\[
\int_{x_1}^{x_2} G_{mn}(s)ds = T - ((n + 1)R - Q + (m + 1)P)S + (m + 1)((n + 1)R - Q)
\]

\[
\int_{x_1}^{x_2} G_{m+1,n}(s)ds = -T + ((n + 1)R - Q + mP)S - m((n + 1)R - Q)
\]

\[
\int_{x_1}^{x_2} G_{mn+1}(s)ds = -T + (nR - Q + (m + 1)P)S - (m + 1)(nR - Q)
\]

\[
\int_{x_1}^{x_2} G_{m+1,n+1}(s)ds = T - (nR - Q + mP)S + m(nR - Q)
\]

When the slope, \(M\), is large, rounding errors can occur and when the line has infinite slope further complications arise. In order to avoid these problems, if the slope is greater than one then \(x\) is taken to be the independent variable in the equation of the line and so the expressions in the parametric equation, (84), are exchanged. Similar expressions for the line integral are then derived in exactly the same way.

B.5.2 Extended Line Integral

Because of the piecewise definition of the continuous goodness function, the expressions for the line integral derived above are only valid when the line is completely contained within one pixel of the image. Lines joining consecutive elements of an ARM pass through a number of pixels and so integrals along them cannot be evaluated directly from (87). The line must be divided into subsections that lie within single pixels and the integrals over these subsections summed to yield the line integral as a whole. The marching squares process used to evaluate the whole line
Appendix B Algorithms

integral is described in this section. The input to the process is the co-ordinates in the image of
the two contour elements, \((x_1, y_1)\) and \((x_2, y_2)\), between which the line integral is required.
- The equation of the line joining the two points is calculated as in (84), the independent
variable being either \(x\) or \(y\) depending on which one minimises the slope.
- Two pairs of running co-ordinates, \((x_{\text{prev}}, y_{\text{prev}})\) and \((x_{\text{next}}, y_{\text{next}})\) are set equal to \((x_1, y_1)\).
- Two update values, \(X_{\text{up}}\) and \(Y_{\text{up}}\) are defined as

\[
X_{\text{up}} = \begin{cases} 
1 & \text{if } x_2 > x_1 \\
-1 & \text{otherwise}
\end{cases}, \quad \text{and } Y_{\text{up}} = \begin{cases} 
1 & \text{if } y_2 > y_1 \\
-1 & \text{otherwise}
\end{cases}
\]

- Initialise the total line integral to zero.
- While \((x_{\text{prev}}, y_{\text{prev}})\) are not equal to \((x_2, y_2)\)
  - Find the point on the line at which \(x = \text{int}(x_{\text{next}}) + X_{\text{up}}\) and the point at which \(y = \text{int}(y_{\text{next}}) + Y_{\text{up}}\). Choose the point closer to \((x_{\text{next}}, y_{\text{next}})\) and set \((x_{\text{next}}, y_{\text{next}})\) equal to it.
  - In case of rounding errors, check that the end of the line has not been overshot. If \(x_{\text{next}} * X_{\text{up}} > x_2\), set \(x_{\text{next}}\) equal to \(x_2\). If \(y_{\text{next}} * Y_{\text{up}} > y_2\), set \(y_{\text{next}}\) equal to \(y_2\).
  - The line joining \((x_{\text{prev}}, y_{\text{prev}})\) and \((x_{\text{next}}, y_{\text{next}})\) is now completely contained within one
pixel and so the line integral between those points can be evaluated using (87).
  - Add the value of the line integral between \((x_{\text{prev}}, y_{\text{prev}})\) and \((x_{\text{next}}, y_{\text{next}})\) to the total
line integral value.

B.5.3 Surface Integrals

In this section and the next, an algorithm for evaluating surface integrals of a bilinearly
interpolated goodness function over triangular regions in an image, which extend over several
pixels, is described. This algorithm is used to approximate the image energy change induced by
a move of a single contour element in the SIG ARM of section 5.3.6.1.3. As is demonstrated in
the next section, these triangular regions can always be decomposed into a set of atomic sub-
regions consisting of whole pixels, sub-pixel rectangular regions and sub-pixel, upright, right-
angled triangular regions. The surface integral, therefore, can also be broken down into the sum
of surface integrals over such regions. In this section expressions for the surface integral of a
bilinearly interpolated function over these atomic regions are derived. To simplify the following
calculation, the expression for the bilinearly interpolated goodness function, (83), is rewritten as

\[
G(x, y) = Ax y + Bx + Cy + D \tag{90}
\]

where

\[
\begin{align*}
A &= G(m, n) - G(m, n + 1) - G(m + 1, n) + G(m + 1, n + 1) \\
B &= (n + 1)G(m, n) - nG(m, n + 1) - (n + 1)G(m + 1, n) + nG(m + 1, n + 1) \\
C &= (m + 1)G(m, n) - (m + 1)G(m, n + 1) - mG(m + 1, n) + mG(m + 1, n + 1) \\
D &= (m + 1)(n + 1)G(m, n) - (m + 1)nG(m, n + 1) - m(n + 1)G(m + 1, n) + mnG(m + 1, n + 1)
\end{align*}
\]
Figure 79 shows examples of the types of atomic region – rectangular and upright, right-angled triangular, over which expressions for the surface integral are required.

Figure 79 The atomic sub-pixel regions over which expressions for the surface integral are required.

**B.5.3.1 Whole pixel regions**

The integral of expression (90) over an entire pixel is simply the average of the four goodness values at the corners:

\[
\int_{y=n}^{n+1} \int_{x=m}^{m+1} G(x,y) \, dx \, dy = \frac{1}{4} (G_l(m,n) + G_l(m+1,n) + G_l(m,n+1) + G_l(m+1,n+1)).
\]

(92)

The proof is omitted, but (92) is simple to verify either directly from (90), or from (93) in the next section.

**B.5.3.2 Sub-pixel rectangular regions**

An expression for the integral over a rectangular sub-region is obtained as follows:

\[
\int_{y_1}^{y_2} \int_{x_1}^{x_2} G(x,y) \, dx \, dy = \int_{y_1}^{y_2} \int_{x_1}^{x_2} (Ax - Bx - Cy + D) \, dx \, dy
\]

(93)

\[
= \int_{y_1}^{y_2} \left( \frac{1}{2} Ay(x_2^2 - x_1^2) - \frac{1}{2} Bx(x_2^2 - x_1^2) - Cy(x_2 - x_1) + D(x_2 - x_1) \right) \, dy
\]

\[
= \frac{1}{4} a [A(x_2 + x_1)(y_2 + y_1) - 2B(x_2 + x_1)(y_2 + y_1) - 2C(y_2 + y_1) + 4D]
\]

where \( a = (x_2 - x_1)(y_2 - y_1) \) is the area of the rectangle.
B.5.3.3 Sub-pixel, upright, right-angled triangular regions

To find a similar expression for the surface integral over right-angled triangular regions aligned with the co-ordinate axes, the equation of the hypotenuse line must first be found as in the line integrals of section B.5.1. Consider first the triangle with vertices \((x_i, y_i), (x_2, y_2),\) and \((x_3, y_3)\), in Figure 79. The equation of the hypotenuse is given by \(y = Mx + L\), where \(M = \frac{y_2 - y_1}{x_2 - x_1}\), \(L = \frac{x_2 y_1 - x_1 y_2}{x_2 - x_1}\), and the integral is then

\[
\iint_{\Delta} G_l(x, y) \, dx \, dy = \int_{x_i, y_i}^{x_2, y_2} (Axy - Bx - Cy + D) \, dy \, dx
\]

\[
= \frac{1}{8} AM^2 \left( x_2^4 - x_1^4 \right) + \frac{1}{6} \left( 2AML - 2BM - CM^2 \right) \left( x_2^3 - x_1^3 \right)
\]

\[
+ \frac{1}{4} \left( A \left( L^2 - y_1^2 \right) - 2B(L - y_1) - 2CML + 2DM \left( x_2^2 - x_1^2 \right) \right)
\]

\[
+ \frac{1}{2} \left( 2D(L - y_1) - C \left( L^2 - y_1^2 \right) \right) (x_1 - x_2)
\]

Slightly different expressions result for triangles in different orientations. For the triangle with vertices \((x_1, y_1), (x_2, y_2),\) and \((x_1, y_2)\),

\[
\iint_{\Delta} G_l(x, y) \, dx \, dy = \int_{x_i, y_i}^{x_2, y_2} (Axy - Bx - Cy + D) \, dy \, dx
\]

\[
= \frac{1}{8} AM^2 \left( x_2^4 - x_1^4 \right) - \frac{1}{6} \left( 2AML - 2BM - CM^2 \right) \left( x_2^3 - x_1^3 \right)
\]

\[
- \frac{1}{4} \left( A \left( L^2 - y_1^2 \right) - 2B(L - y_1) - 2CML + 2DM \left( x_2^2 - x_1^2 \right) \right)
\]

\[
- \frac{1}{2} \left( 2D(L - y_1) - C \left( L^2 - y_1^2 \right) \right) (x_1 - x_2)
\]

For the triangle with vertices \((x_1, y_2), (x_1, y_1),\) and \((x_2, y_1)\) the slope and intercept are given by \(M = \frac{y_1 - y_2}{x_2 - x_1}\), \(L = \frac{x_2 y_2 - x_1 y_1}{x_2 - x_1}\), and the surface integrals are then given by the same expressions, found in equations (94) and (95) respectively.

B.5.4 Extended Surface Integral

This section explains how the expressions derived above can be used to calculate the surface integral of the bilinearly interpolated goodness function over the extended triangular regions that result when the position of a single contour element is updated. For this application of the algorithm, the set of triangles over which the surface integral will be required can be constrained considerably. The contour that uses this algorithm uses a greedy energy minimisation procedure
and so the contour element positions are all integer pixel values. In non-trivial cases, i.e., the contour element moves, the new position of an element always lies in the 3x3 neighbourhood of the current position. Thus, two of the vertices of the triangle are always corners of the same pixel, which means that the triangle can never completely contain any full pixel. The third vertex cannot be a corner of the same pixel, as consecutive elements cannot be within less than three pixels of each other, see section 4.5.1.1.

The algorithm proceeds by examining each pixel through which each side of the triangle passes and evaluating the integral over the intersection. Rather than finding the integral over that intersection directly, the integral over the whole square is evaluated and then the integral over the region of that square that is excluded by the edge currently being considered is evaluated and subtracted. This avoids complications that might otherwise occur when two edges intersect the same square. Figure 80 shows a typical triangle over which the surface integral of the goodness function might be required. The surface integral is evaluated over the green area and the goodness is also summed over each pixel intersected by the triangle. The former quantity is then subtracted from the latter to obtain the surface integral over the triangular region.

B.5.4.1 Main Algorithm

- Identify the vertex with the lowest y-co-ordinate (highest in the image) and order the vertices in a clockwise manner. If two vertices have equally low y-co-ordinate, one can be chosen arbitrarily. In Figure 80, the vertices are ordered, as labelled, 1-2-3.
- Initialise an empty list to keep track of the pixels that have been visited.
- Initialise the value of the integral to zero.
- Traverse the edges in order, using the marching squares process described in section B.5.2. For each pixel passed through,
Appendix B Algorithms

- Check whether the square has been visited and if not add the surface integral over that square, given by (92), to the total.
- Find the value of the surface integral over the part of the pixel that is not part of the triangle and subtract it from the total. There are sixteen ways in which an edge can intersect a pixel and each case is shown in Figure 81. In that figure, the region outside the triangle is highlighted, in green, for each type of intersection. The figure further shows how each such region is decomposed into atomic regions over which the surface integral can be evaluated using the results in the previous section, and also, how each type of intersection is identified. The co-ordinates of each vertex of each atomic region are readily available from the marching squares process.

In less constrained circumstances, when the vertices of the triangle are non-integer, and may be arbitrarily close together or far apart, many more intersection cases arise, particularly in pixels that contain vertices, and the procedure becomes considerably more complex. No such algorithm is described here. Even in this constrained scenario, however, great care must be taken to avoid small rounding errors, which can cause large errors in the overall surface integral and lead to non-convergence of the ARM.
Appendix B Algorithms

Figure 81 The sixteen ways in which one of the edges of the triangle can intersect a consistent square.
Appendix C Receiver Operating Characteristic Analysis.

Appendix C Receiver Operating Characteristic Analysis.

Receiver operating characteristic (ROC) analysis is used in signal detection as a means of estimating the degree to which criteria can differentiate between inputs consisting of noise alone and those containing some kind of signal distorted by that same noise. It has been used extensively in psychophysical research, in a similar way, to investigate the ability of human test subjects to identify noisy signals from noise alone, e.g., [EBT93]. Many applications also occur within the medical community, where ROC analysis is often used as a means of evaluating the discriminatory power of different clinical techniques, see for example [RF92]. Good introductory reviews of the subject, both with a leaning toward medical applications, in particular medical image analysis, have been provided by Swets, [Swe79], and Metz, [Met78]. A more recent overview of its application can be found in [Met92].

C.1 The ROC curve

The signal detection problem is a binary decision making problem and this is the type of problem to which ROC analysis is applied. In a signal detection problem, two types of input case are encountered: positives, in which there is a signal present, and negatives, which consist entirely of noise. A binary decision making system classifies each case as either positive or negative, which gives rise to four possible decision outcomes. If a case is correctly classified as positive, it is referred to as a true positive; if it is correctly classified as negative, it is a true negative. A case that is classified as positive, but is in fact negative, is called a false positive; and cases that are incorrectly classified as negative are known as false negatives. If it is assumed that, within some set of test cases, the true classification of each case is known, then the actual numbers of positive and negative cases are known. Two identities then exist:

\[
\text{Actual No. Positives} = \text{No. True Positives} + \text{No. False Negatives}
\]

\[
\text{Actual No. Negatives} = \text{No. True Negatives} + \text{No. False Positives}
\]

The performance of a system over a particular set of data can then be described by independent counts of just two of the four possible decision outcomes, for example, the number of true positives and the number of false positives. The probability of occurrence of each of the four decision outcomes can be found easily by taking the ratio with the total number of cases. Of more interest for assessing the performance of the system, however, are the ratios of each count to the total number of cases of the same type. For example, the true positive ratio is the ratio of the number of true positives to the actual number of positives and the false positive ratio is the ratio of the number of false positives to the actual number of negatives.

Most non-trivial decision making systems contain parameters, such as decision thresholds, whose settings can affect the strictness of the decisions that are made and hence change the number of times each decision outcome occurs. The ROC curve of a system is the
Appendix C Receiver Operating Characteristic Analysis.

curve that is obtained by plotting two independent ratios, for example, the true and false positive ratios, against each other, while such a parameter is varied. In general it is not possible to find the true and false positive ratios produced by a system for every possible setting of the internal parameter, which may be real valued. A finite number of operating points, which correspond to settings for the internal parameter, must be identified and the ratios are calculated at each. A number of points on the ROC curve are thus obtained, which can be interpolated to approximate the true curve.

![Figure 82 Examples of ROC curves.](image)

Figure 82 shows three ROC curves, which are comprised of a finite number of points that have been interpolated linearly. The blue curve represents the type of performance that might be expected from a real world system. As the number of true positives increases, so does the number of false positives, but parameter settings can be found such that the true positive ratio is much greater than the false positive ratio, so that a reasonable degree of discrimination between the two types of case is provided. For an ideal system, it is possible to find a setting for the internal parameter such that every case is classified correctly and so the true positive ratio is one while the false positive ratio is zero. The pink ROC curve in Figure 82 represents the performance of such a system. At some lower operating points, the true positive ratio may be less than one, as shown, and at some higher ones, the false positive ratio may become greater than zero. Finally, the green curve lies along the diagonal and characterises the performance of a system whose classification power is equivalent to random chance. This diagonal is known as the “chance diagonal”. It is always assumed that the two extreme parameter settings exist, one at which every case is classified negatively and one where every case is classified positively, so that the curve always passes through the points (0, 0) and (1, 1).
Appendix C Receiver Operating Characteristic Analysis.

A higher degree of discrimination between the two types of case is generally represented by an ROC curve that is closer to the curve obtained from a system with perfect performance – shown in pink in Figure 82. It is often possible to determine which of two systems provides the highest overall performance simply by inspection of their ROCs. If one curve is consistently closer to the ideal curve then its performance is better overall. As an ROC curve approaches the ideal curve, the area underneath the curve increases and this area is often used, [Swe79, MH84], as an overall measure of the classification performance provided by a system.

In many applications of ROC, particularly medical applications, [Met78, Swe79], the system being assessed is a human. For example, in medical imaging, often of interest is the extent with which a particular type of data provides an expert the ability to distinguish affliction from non-affliction in the patient from which the image was obtained. As it is hard to explicitly tune parameters in such a system, a rating scale is generally employed. The expert is asked to express the degree of certainty with which a decision can be made, for example, if a five point rating scale is used, a rating of 1 might express certainty of affliction and a rating of 5 certainty of non-affliction. A binary decision about each case can then be made from the rating scale by thresholding at each rating, e.g., every patient rated above 3 is assumed to be healthy, and so a point on an ROC curve can be obtained from each rating.

C.2 System Model

It is very common to adopt a system model for use in conjunction with ROC analysis and that model is described in this section. In the model it is assumed that a binary decision making system determines each decision by making a one-dimensional measure on each case and applying a threshold to the value of that measure. For a particular set of cases there is then a distribution of values corresponding to each type of case, i.e., there is a distribution of values of the measurement that arise for negative cases and an independent distribution of values that arise from positive cases. This model is depicted in Figure 83, which shows how each type of decision outcome can arise by application of a threshold in this way. In that figure, “TP”, “FP”, “TN”, FN”, stand for true positive, false positive, true negative and false negative, respectively, and indicate the region of the distribution where cases that give rise to these decision outcomes reside. The position of the decision threshold corresponds to the value of the internal system parameter that controls the strictness of the decisions, or the operating point. From Figure 83 it can clearly be seen how varying the position of the decision threshold alters the counts of each type of decision outcome. For example, if the value is increased (moved to the right in the figure), the number of false positives decreases, but so does the number of true positives.
Appendix C Receiver Operating Characteristic Analysis.

Typically, the distributions overlap somewhat so that at any non-extreme position of the decision threshold, there is a trade off between true and false positives. For an ideal system however, the two distributions are completely separated so that the decision threshold can be placed completely between them, which gives rise to an ROC curve such as that shown in pink in Figure 82. Similarly, when the probability distributions are identical, the true and false positive rates are always equal so giving rise to the ROC curve shown in green in the same figure.

![Figure 83 Depicts the distributions of measurements made by a system on a set of test cases and how the application of a decision threshold gives rise to each type of decision outcome.](image)

In Figure 83 the two distributions are depicted as Gaussian distributions. Although, this Gaussian assumption is a common extension to the system model (as described in the next section), the distributions may in fact take any form. Whatever form they take, so long as they are valid probability distributions, the ROC curve that arises from a system that fits the model must be monotonic, because the probability density functions are strictly non-negative.

Golfarelli, et al, [GMM97], recently looked at a very similar type of curve, which they use to assess the performance of biometric verification systems. Such systems attempt to verify that a given person is who they say they are by comparison of measurements taken from that person, for example, texture measures taken from an image of the iris, with similar measurements stored in a database. After such a comparison, the hypothesis that this person is who they claim to be can be accepted or rejected depending on how similar the new measurements are to those stored in the database. A parameter controls how similar the measurements have to be in order for the hypothesis to be accepted. By tuning this parameter the numbers of false acceptances and false rejections can be varied in a very similar way to the
number of false positives and false negatives in the binary decision making systems discussed above. Golfarelli, et al, plot false acceptance rate against false rejection rate in order to characterise the performance of their systems. By assuming a similar model, based on general (non-specific) probability distributions, they show that the resulting curves must not only be monotonic, but must also be concave.

Their proof, however, does not apply to the two class case of interest here and, in general, an ROC curve constructed by plotting true and false positive rates need not be convex (which would be implied by concavity of the corresponding false positive against false negative curve). There are potentially an infinite number of classes in the problem investigated in [GMM97], and at one stage (equation (9) in [GMM97]) they show that the false acceptance rate is equal to the integral of the probability density function for actual measurement occurrences over the acceptance region. The acceptance region in the system model used here corresponds to the set of cases whose measurement value is greater than the decision threshold. The probability density function for the actual occurrence of a particular system measurement value is a Bayesian combination of the two density functions shown in Figure 83. However, the false positive rate (c.f. false acceptance rate) is the integral of just the probability density function for the negative cases over this region rather than that of the combined density function. This stage therefore does not apply to the problem being investigated here and the subsequent proof of concavity breaks down.

![Figure 84 An example showing how a non-convex ROC curve can arise.](image)

Several examples of non-convex ROC curves can be found in the results presented in Chapter 7, for example the ROC curve produced by the ARM using Zhu and Yuille's chromaticity model over the clothes test set, shown in Figure 50. The problem with this chromaticity model is that many dark pixels that are not part of the region of interest are classified positively even at very low settings for the SCM tolerance parameter, k, which define
Appendix C Receiver Operating Characteristic Analysis.

the operating points. These pixels cause the distribution of values assigned by the decision making system to the negative cases to have an extra peak at a very high value of the measurement. Figure 84 shows an example of two probability distributions that cause the ROC curve to be non-convex in a similar way to the curve shown in Figure 50. The distribution for positive cases is Gaussian, but the distribution for the negative cases is a bimodal Gaussian distribution and, as shown, the resulting ROC curve, which is calculated directly from the expressions for these probability distributions, is clearly non-convex.

C.3 Performance Measures from ROC curves.

As mentioned in the previous section, the area under the curve intuitively appears to be a reasonable measure of overall system performance that can be derived from an ROC curve. This area is often used in exactly this way, as suggested in [MH84, Swe79], for comparison of decision making systems. If the underlying distributions are assumed to be Gaussian, then the ROC curve takes a particular form and is known as a binormal ROC curve. The area, A, underneath the curve can then be shown to be a monotonic function of the difference between the means of the two distributions taking into account both variances and so gives a good indication of the degree of discrimination that is provided. Suppose the two underlying normal distributions are \( N(\mu_1, \sigma^2_1) \) and \( N(\mu_2, \sigma^2_2) \), then

\[
A(z(A)) = \frac{(\mu_1 - \mu_2)}{(\sigma^2_1 + \sigma^2_2)^{1/2}}.
\]

where \( z \) is the inverse cumulative normal distribution function, with zero mean and unit variance.

The area under an ROC curve can be computed by interpolating linearly between consecutive points and using a trapezium rule, [MH84], but if this binormal assumption is made, the parameters of the binormal model can be fit to the discretely defined curve by maximum likelihood estimation. Dorfman and Alf, [DA69], developed such a maximum likelihood procedure, which was later implemented with some extensions by Metz, et al, [MKW85]. Metz’ program, which is freely available, also provides estimates of the goodness of fit of a binormal model to a discrete ROC curve and estimates of \( A, z(A) \), and various other indices derived from the curve together with variances and covariances of each that indicate the level to which each measure can be relied upon.

Binormal ROC curves follow a straight line when they are plotted on binormal coordinates, [Swe79]. The slope and intercept of this straight line, customarily known in the literature as \( b \) and \( a \) respectively (although \( b \) is sometimes used to denote the reciprocal of the slope as in [DA69]), are two particularly useful values that are provided by the Dorfman and Alf procedure. The values of \( a \) and \( b \) may be expressed as:
Appendix C Receiver Operating Characteristic Analysis.

\[ b = \frac{\sigma_1}{\sigma_2}, \quad \text{and} \quad a = \frac{(\mu_1 - \mu_2)}{\sigma_2}. \]  

(98)

and are used in the program to calculate the value of z(A). Unlike z(A), which is useful because it provides a uni-dimensional measure of performance, a and b, completely express a unique binormal ROC curve.

### C.4 Operating Point Selection

An advantage of ROC analysis is that the performance is assessed independently of any particular operating point. This is useful as the exact values and losses associated with particular decision outcomes may not be known at the early stages of system development. Once this information does become available, it can be used to select the most appropriate operating point directly from the ROC curve. A common index is

\[ \beta = \frac{P(n)}{P(p)} \cdot \frac{V_{TN} + C_{FP}}{V_{TP} + C_{FN}} \]  

(99)

where V and C represent the application specific, user defined, values and costs of the decision outcomes denoted by their subscripts, P(p) is the overall probability of a positive case occurring and P(n) that of a negative case. \( \beta \) describes a gradient on the ROC curve and the point at which that gradient occurs corresponds to the optimum operating point for that application. The point on the ROC curve can be found quite easily, for example by using a Newton method starting with some small, but non-zero, value for the false positive ratio. No such procedure has been implemented and tested during the extent of this project however.

Alternatively, the Metz' implementation of the maximum likelihood estimation program calculates a list of true positive ratios for user specified false positive ratios together with 95% confidence intervals for those fractions. Thus if the false positive ratio tolerance is known a priori, an estimate for the true positive ratio can be found. A list of false positive ratios for set true positive ratios may also be calculated in case the requirement for true positives is known a priori rather than the tolerance for false positives.

### C.5 Problems

The freeware implementation of the Dorfman and Alf maximum likelihood estimation procedure, due to Charles Metz, [MKW85], is designed for medical applications where relatively small amounts of rating scale data is collected by hand from medical experts. For the applications in this thesis and analysis of automatic systems in general, the implementation poses a number of problems:

- The user is limited to a fixed number (eleven) of operating points to which the curve is fitted. This is reasonable for data generated by hand but when the data is generated
automatically, for greater accuracy in the curve fitting procedure, as many operating points should be used as possible.

- The maximum likelihood estimation procedure is based on an initial estimate of the binormal curve parameters to which a Newton method is applied to obtain accurate estimates. Non-convergence of this Newton method has proved fairly common when applied to the ROC curves generated here. Although Metz claims that non-convergence is rare for rating scale data, a failure rate of approximately 40% has been observed for these curves. Because of these problems, for the present a trapezium rule estimation of the area under ROC curves is used rather than the Dorfman and Alf technique. This measure is still fairly good, and in fact has the advantage that the binormal assumption no longer has to be made. More care must be taken, however, to ensure that the operating points always provide a good spread along the entirety of the curve. In [Met92], Metz reports some experiments in which areas under some ROC curves were estimated using the Dorfman and Alf procedure and a simple trapezium rule and the results were found to often agree, but that those estimated using the former method were "somewhat more reliable". Furthermore, the measures of uncertainty that are obtained when the maximum likelihood procedure is used are no longer available.
Appendix D Further Results

This appendix contains plots of the performance measure, $z(A)$, obtained during the training phase of the SCM comparisons that were performed using the DAG ARM in section 7.4.1.

D.1 Clothes Training Set, Intensity Model Comparison
Appendix D Further Results

D.2 Clothes Training Set, Chromaticity Model Comparison

Bingham, linear downweighting.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=3$</td>
<td>2.20241</td>
<td>2.1937</td>
<td>2.19792</td>
<td>2.19701</td>
</tr>
<tr>
<td>$p=5$</td>
<td>2.21965</td>
<td>2.19403</td>
<td>2.20838</td>
<td>2.21723</td>
</tr>
<tr>
<td>$p=7$</td>
<td>2.18555</td>
<td>2.19338</td>
<td>2.19124</td>
<td>2.14315</td>
</tr>
</tbody>
</table>

Bingham, sq. root downweighting.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=3$</td>
<td>2.21929</td>
<td>2.22162</td>
<td>2.08029</td>
<td>2.13644</td>
</tr>
<tr>
<td>$p=5$</td>
<td>2.23183</td>
<td>2.23509</td>
<td>2.24511</td>
<td>2.15042</td>
</tr>
<tr>
<td>$p=7$</td>
<td>2.17651</td>
<td>2.17972</td>
<td>2.22802</td>
<td>2.20034</td>
</tr>
</tbody>
</table>

Healey's model

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=3$</td>
<td>2.22169</td>
<td>2.18747</td>
<td>2.23986</td>
<td>2.23183</td>
</tr>
<tr>
<td>$p=5$</td>
<td>2.13547</td>
<td>2.16698</td>
<td>2.17167</td>
<td>2.19903</td>
</tr>
<tr>
<td>$p=7$</td>
<td>2.17651</td>
<td>2.17072</td>
<td>2.22802</td>
<td>2.20034</td>
</tr>
</tbody>
</table>

Angular Normal

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p=3$</td>
<td>2.06246</td>
<td>2.17977</td>
<td>2.06623</td>
<td>2.07338</td>
</tr>
<tr>
<td>$p=5$</td>
<td>2.16131</td>
<td>2.09253</td>
<td>2.15947</td>
<td>2.19516</td>
</tr>
<tr>
<td>$p=7$</td>
<td>2.14876</td>
<td>2.17332</td>
<td>2.14005</td>
<td>2.15997</td>
</tr>
</tbody>
</table>
Appendix D Further Results

Gaussian Normalised Colour

<table>
<thead>
<tr>
<th>p = 3</th>
<th>2.1232</th>
<th>1.99336</th>
<th>1.9863</th>
<th>2.11056</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 5</td>
<td>2.01404</td>
<td>1.98643</td>
<td>2.01646</td>
<td>2.05545</td>
</tr>
<tr>
<td>p = 7</td>
<td>1.86272</td>
<td>2.09204</td>
<td>2.03454</td>
<td>2.0782</td>
</tr>
</tbody>
</table>

Oriented Planar Gaussian

<table>
<thead>
<tr>
<th>p = 3</th>
<th>2.03905</th>
<th>2.05144</th>
<th>2.11338</th>
<th>2.03608</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 5</td>
<td>2.08289</td>
<td>2.154</td>
<td>2.08374</td>
<td>2.11098</td>
</tr>
<tr>
<td>p = 7</td>
<td>2.12698</td>
<td>2.08159</td>
<td>2.09517</td>
<td>2.06373</td>
</tr>
</tbody>
</table>

Tri-variate Gaussian model

<table>
<thead>
<tr>
<th>p = 3</th>
<th>2.04901</th>
<th>2.08527</th>
<th>2.13058</th>
<th>2.07658</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 5</td>
<td>2.09186</td>
<td>2.09432</td>
<td>2.09199</td>
<td>2.09511</td>
</tr>
<tr>
<td>p = 7</td>
<td>2.09792</td>
<td>2.1039</td>
<td>2.12959</td>
<td>2.09537</td>
</tr>
</tbody>
</table>

Zhu and Yuille's model

<table>
<thead>
<tr>
<th>p = 3</th>
<th>1.10345</th>
<th>1.07798</th>
<th>1.08171</th>
<th>1.08414</th>
</tr>
</thead>
<tbody>
<tr>
<td>p = 5</td>
<td>1.07871</td>
<td>1.08458</td>
<td>1.10017</td>
<td>1.09453</td>
</tr>
<tr>
<td>p = 7</td>
<td>1.04191</td>
<td>1.05336</td>
<td>1.0576</td>
<td>1.05974</td>
</tr>
</tbody>
</table>

242
Appendix D Further Results

D.3 Tracks Training Set, Intensity Model Comparison

Uniform intensity

<table>
<thead>
<tr>
<th>β</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>r = 3</td>
<td>3.31133</td>
<td>3.3946</td>
<td>3.2744</td>
<td>3.2552</td>
</tr>
<tr>
<td>r = 5</td>
<td>3.31133</td>
<td>3.30723</td>
<td>3.30723</td>
<td>3.30723</td>
</tr>
<tr>
<td>r = 7</td>
<td>3.32819</td>
<td>3.32819</td>
<td>3.32819</td>
<td>3.32819</td>
</tr>
</tbody>
</table>

Gaussian Intensity

<table>
<thead>
<tr>
<th>β</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>r = 3</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
</tr>
<tr>
<td>r = 5</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
</tr>
<tr>
<td>r = 7</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
<td>3.3873</td>
</tr>
</tbody>
</table>

Planar surface intensity

<table>
<thead>
<tr>
<th>β</th>
<th>0.05</th>
<th>0.1</th>
<th>0.3</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>r = 3</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
</tr>
<tr>
<td>r = 5</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
</tr>
<tr>
<td>r = 7</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
<td>3.35464</td>
</tr>
</tbody>
</table>
Appendix D Further Results

D.4 Tracks Training Set, Chromaticity Model Comparison

Planar surface intensity, Bingham chromaticity, lin. down-weighting

<table>
<thead>
<tr>
<th>p = 3</th>
<th>p = 3</th>
<th>p = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.39271</td>
<td>3.36497</td>
<td>3.38534</td>
</tr>
<tr>
<td>3.38534</td>
<td>3.34991</td>
<td></td>
</tr>
</tbody>
</table>

Planar surface intensity, Bingham chromaticity, sq. rt. down-weighting

<table>
<thead>
<tr>
<th>p = 3</th>
<th>p = 3</th>
<th>p = 5</th>
<th>p = 3</th>
<th>p = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.37896</td>
<td>3.36289</td>
<td>3.33356</td>
<td>3.3653</td>
<td></td>
</tr>
<tr>
<td>3.33356</td>
<td>3.37011</td>
<td>3.34327</td>
<td>3.29341</td>
<td></td>
</tr>
<tr>
<td>3.39796</td>
<td>3.39837</td>
<td>3.42473</td>
<td>3.34052</td>
<td></td>
</tr>
<tr>
<td>3.42473</td>
<td>3.44306</td>
<td>3.3972</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Healey’s model

Zhu and Yuille’s model

<table>
<thead>
<tr>
<th>p = 3</th>
<th>p = 5</th>
<th>p = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.41877</td>
<td>3.39037</td>
<td>3.42473</td>
</tr>
<tr>
<td>3.39037</td>
<td>3.34052</td>
<td></td>
</tr>
<tr>
<td>3.34052</td>
<td>3.3972</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>p = 3</th>
<th>p = 5</th>
<th>p = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.22395</td>
<td>3.22951</td>
<td>3.29952</td>
</tr>
<tr>
<td>3.22951</td>
<td>3.35991</td>
<td></td>
</tr>
<tr>
<td>3.35991</td>
<td>3.3972</td>
<td></td>
</tr>
</tbody>
</table>

244