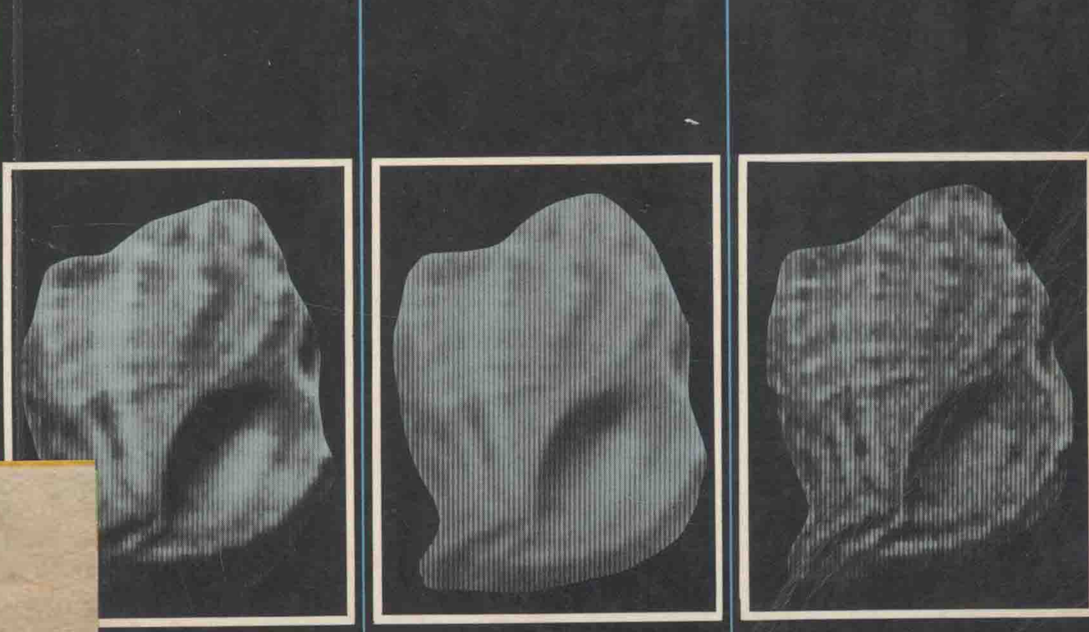
PATTERN RECOGNITION

MIKE JAMES



Pattern Recognition

Mike James

BSP PROFESSIONAL BOOKS
OXFORD LONDON EDINBURGH
BOSTON PALO ALTO MELBOURNE

First published 1987

British Library
Cataloguing in Publication Data

James, Mike
Pattern recognition.
1. Pattern perception
I. Title
006.4 0327

ISBN 0-632-01885-2

BSP Professional Books
Editorial offices:
Osney Mead, Oxford OX2 0EL
(*Orders*: Tel. 0865 240201)
8 John Street, London WC1N 2ES
23 Ainslie Place, Edinburgh EH3 6AJ
52 Beacon Street, Boston
Massachusetts 02108, USA
667 Lytton Avenue, Palo Alto
California 94301, USA
107 Barry Street, Carlton
Victoria 3053, Australia

Set by V & M Graphics, Aylesbury, Bucks
Printed and bound in Great Britain by
Billing & Sons Limited, Worcester.

Preface

Our expectations of what computers can or should be able to do increase daily, and have already reached the point where it seems incredible that there are no walking, talking and seeing computers. In reality we can just about manage to produce the walking and talking computer, but the automation of vision is still a very difficult problem! This is not to say that no progress has been made, and character readers and similar devices are now reasonably common, but there is still a long way to go.

This book is an introduction to pattern recognition as applied to images and image processing and as used in pattern recognition, suitable for a one- or two-term course at undergraduate level. The topics covered are developed as logically as possible in a subject that is essentially a mixture of abstract theory and pragmatic solution, and wherever possible I have tried to emphasise the practical. As low cost image input devices are now available for personal computers such as the IBM PC, listings of BASIC subroutines have been included. The choice of BASIC rather than a more traditional language such as FORTRAN, or a more academic language such as Pascal, is partly due to the availablity of a low cost BASIC complier for the PC (Microsoft's Quick BASIC), and partly due to the simplicity of the language. If you prefer another language or another dialect of BASIC, the programs are very easy to convert. BASIC is like FORTRAN and the particular BASIC style used makes it look very like Pascal, or indeed like any block-structured lanaguage. Some suggestions for short practical projects based on the use and extension of the subroutines can be found at the end of the book.

Wherever possible I have tried to explain the methods in terms that are suitable for implementation on a parallel computer. These parallel algorithms do not always result in the most efficient programs when used with a standard serial computer, and as a result most of the subroutines listed could be replaced by faster versions based on serial methods. The reason for the emphasis on parallel processing is that it is the method that we would like to use if only the necessary hardware were cheaper! With the falling cost of hardware, it seems better to be prepared for parallel thought rather than confining ourselves to serial thinking.

The images used in this book were all prepared using an IBM PC and a commercially available frame grabber – the IMAGE 3C Frame Store, kindly lent us by Dave Hurst of Eltime Ltd, Unit D29, Maldon Industrial Estate, Fullbridge, Maldon, Essex (Tel: 0621 59500).

The programs were written in collaboration with Kay Ewbank and my grateful thanks are due to her. I would also like to thank Professor M. J. B. Duff of University College, London, and my ex-colleagues at the Image Processing Group, for my training in parallel image processing. Their CLIP series is still the most powerful range of parallel computers for image processing. Thanks are also due to my wife Sue for much of the research necessary to complete this manuscript, and to my editor Bernard Watson of BSP Professional Books for his patient encouragement throughout the project.

Mike James

A Note on the Programs

All of the programs listed in this book were written to be complied using Microsoft's Quick BASIC complier. All that is needed to use them is a few extra subroutines to perform image input and display. These are not included because they depend too much on the type of image acquisition system used. The subroutines could be converted to work with an interpreter such as BASICA or GWBASIC but this would severely limit the size of images that could be processed in a reasonable amount of time. An IBM format disk containing all the subroutines listed and others is available. Details can be obtained from Saturn Workshops, Askrigg, Leyburn, North Yorkshire DL8 3LB.

Contents

Chapter One
Pattern Recognition and Images

Giving a computer or a robot the ability to see, even in a very limited sense, is an important and exciting endeavour. In the past, the limitations of computer hardware, both in cost and size, restricted us to trying very simple techniques. Now computer hardware is rapidly becoming the least of our worries. Even a desk-top personal computer, with a few extras, can be used to try out most of the ideas of pattern recognition and image processing. Processing time for images of reasonable quality is usually a few minutes and this means that a specialised high speed computer is still required for a real time system. With the introduction of suitable LSI chips, even these special-purpose image processing computers are bound to become more accessible.

Pattern recognition is a general subject in the sense that patterns take many forms, from the shock waves recorded by a seismograph to the acoustic waves that constitute human speech. However, in this book, the main subject will be the recognition of visual patterns or images. This connection between pattern recognition and image has led to much overlap with *digital image processing* –the general manipulation of images using a computer. This book is mainly about pattern recognition applied to images, but many of the topics covered would not be out of place in a book on image processing. In many cases the methods of the two subjects are the same; it is just the purpose to which they are put which differs.

Images – pictorial patterns – are so common that it is not surprising that much of *pattern recognition* is concerned with them. To *recognise* an image is a term that means many different things to different people, so it would be unreasonable to expect a single coherent theory to emerge. If such a theory existed, then many of the problems in the wider field of artificial intelligence would also be solved, and pattern recognition devices would be commonplace. This is not to say that pattern recognition has yielded no practical results; it is rather that a variety of approaches have been used.

There is a tendency for workers to invent their own approaches and theories in response to the particular practical problem with which they are confronted. On the other hand, in the absence of any confrontations with real problems, there is a great temptation to theorise using abstract mathematics. Such theories are usually all-embracing, but too general or difficult to be of much use. It is possible to pursue pattern recognition as a branch of abstract mathematics; it is also possible to pursue it as a theory-less branch of pragmatic engineering. The best approach is somewhere in-between. There is a great deal of theory that can be used to guide the exploration and eventual solution of a practical problem, but it is far from complete.

One of the difficulties of explaining the underlying theories is that pattern recognition often uses results from other subjects that are relatively easy to state but very hard to prove without relying on the main body of the subject's theory. Students of pattern recognition and image processing will benefit from knowledge of probability and statistics, engineering mathematics such as transforms and differential equations, computer science and diverse subjects such as psychology, photography, photogrammetry, computer graphics, electronics and space science. How much needs to be known from each of these areas depends on the exact application in hand, but the need to track down background information from other subjects is a constant requirement of most pattern recognition and image processing.

In this book the major ideas have been presented in such a way that their relevance, both theoretical and practical, to pattern recognition/image processing should be clear, but should more detail be required then additional reading will be called for. For example, in Chapter Four the theory and practice of the Fourier transform, filtering and the fast Fourier transform algorithm are described in sufficient detail for their workings to be understood, and to show how they apply to pattern recognition/image processing – but this is just a special application of the wider subject of digital signal processing. If you need more information or background reading then one of the suggestions in the Further Reading section at the end of this book should prove useful.

Image processing

The recognition of images obviously involves the use of a certain amount of special computer hardware to acquire and display images. The ability to process images using a computer is not only useful for pattern recognition; the general area of image processing covers computer graphics, image enhancement, image restoration, image analysis, image compression and even special video effects for television. The fact that there is much overlap in the hardware used for these different purposes has led to a certain amount of overlap in methods. Many of the techniques described in this book are found in other branches of image processing, doing slightly different jobs. This can be the cause of much confusion, so it is important to understand exactly why a particular technique is likely to help with the task of image recognition, as opposed to, say, image enhancement. On the other hand, some of the methods and applications of other areas of image processing are useful in the early stages of image recognition. For example, image enhancement and image restoration are attempts to improve the quality of an image that has been degraded in some way, and improving the image quality might make it easier to recognise.

Colour images

A two-dimensional image is an arrangement of colours within a finite border. We can simplify things a little by considering only monochrome images, as a full-colour image can be regarded as a mixture of three monochrome images – one showing the red content, one showing the green content and one showing the blue content. This three-colour representation of a full colour image is possible because of the way that human colour perception works. Given any colour it is possible to find a mixture of red, green and blue light that reproduces it. In this sense, colour vision is a three-dimensional process and any colour can be defined as a triple of numbers (r,g,b) that give the amount of red, green and blue light that have to be mixed to produce that colour. The way that these three colours mix can be seen in Fig. 1.1 which is usually referred to as a *colour cube*. The use of red, green and blue is not essential and any three independent colours (i.e. no one of them can be produced from the remaining two) can be used. For the purposes of pattern recognition, and image processing in general, the combination of red, green and blue is a good choice because these are the colours used in a standard colour video display. Thus a colour camera can be made to supply three signals corresponding to the red, green and blue components of the image, and a colour display will reconstruct a full colour screen when supplied with these signals.

Although it can be proved that three monochrome images can contain the same amount of information as a colour image, it is a matter of observation that a single monochrome image can contain all of the important information in a colour image. For example, it seems obvious that a well-known person's face will be as recognisable in a black and white photograph as in a colour photograph. This may be obvious but it is not something that can be proved. However, in nearly all practical cases a monochrome image will suffice and, as most computers are hard pushed to process such an image in a reasonable amount of time, most of pattern recognition theory ignores colour. The fact that a colour image could be processed as three separate monochrome images is often quoted as a justification for this approach, but there is no guarantee that colour processing will not require something extra. The point is that a

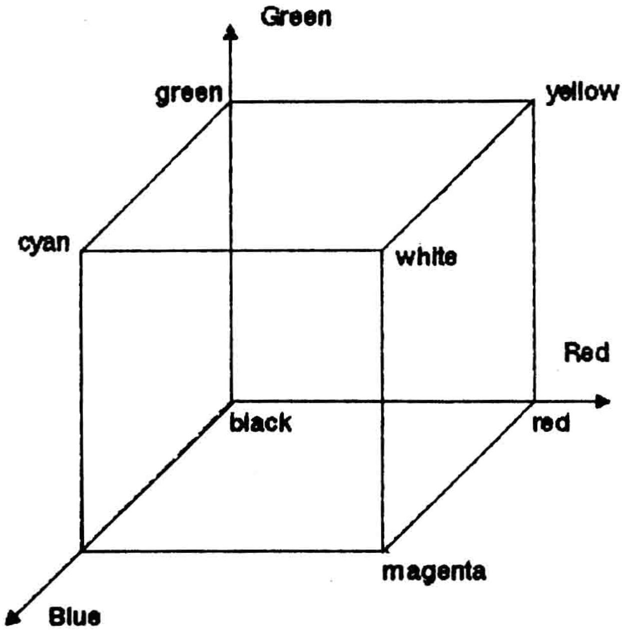


Fig. 1.1 A colour cube

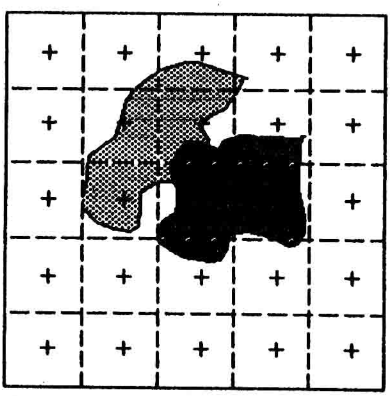
colour image is not three independent monochrome images, but three highly related images and processing them separately may lose important information. The real justification for concentrating on monochrome processing is simply the observation that most of the information in an image can be represented adequately using monochrome.

Grey level images

Another name for a black and white monochrome image is a grey level image because each point in the image is assigned a single number that indicates how *bright* or *grey* it is. Mathematically a grey level image can be represented by a function of two variables f(x,y) which gives a number a, z=f(x,y), that corresponds to the grey level at the point x,y. As an image is usually only defined within a finite region, the function that represents it can be assumed to be non-zero only within a bounded region. By considering the physical characteristics of images you can be more specific about the type of functions that are needed to represent them. For example, as in any real image there will be a brightest and a darkest point the functions must be bounded – i.e. lie between some minimum and maximum value. In the same way, they can also be assumed to be non-negative, and integrable. There is much more that can be made of the relationship between images and functions and it is tempting to see the study of images as a branch of the theory of functions but, as will be explained later, just because something is represented by a mathematical object it is not necessary for all of the properties of that object to be relevant and vice versa. In particular, there are many operations that apply to functions when they are regarded as images, and even seem natural and obvious, but which are most unnatural as a part of the general function theory. For example, forming the histogram of brightness values is natural and obvious for an image but not in the general theory of functions. This does not mean that a knowledge of the theory of functions is not useful, in particular it is worth knowing about the elementary calculus of a function of two variables and about integral transforms such as the Fourier transform. These topics are introduced briefly as they apply to images in later chapters.

Digitisation

Another reason why the theory of functions is not entirely appropriate is that computer processing images are not represented as functions but as discrete arrays of numbers. In this respect practical images are more like matrices than functions. The process of converting an image into an array of numbers is referred to as digitisation. Although at first sight digitisation is a simple process it can be quite tricky. There are two components to consider – *spatial quantisation* and *grey level* or *luminance quantisation.*



0	0	0	0	0
0	4	4	0	0
0	4	7	7	0
0	0	0	0	0
0	0	0	0	0

+ represents a sampling point

Fig. 1.2 Spatial quantisation of an image

Spatial quantisation

Spatial quantisation corresponds to sampling the brightness of the image at a number of points, usually a rectangular grid. This spatial quantisation gives rise to an array of numbers A which can be taken to be an approximation to the original image f(x,y). Each element of the array a_{ij} is referred to as a *pixel* which stands for picture element. Figure 1.2 illustrates a simple sampling scheme.

This of course raises the question of how well A approximates f(x,y). If n^2 samples are taken at regular intervals within a bounding square then it is obvious that the approximation improves as n increases. Roughly speaking, as long as enough samples are taken, a spatially quantised image is as good as the original image. More precisely the original image can be reconstructed exactly from the digitised image as long as the sampling frequency (in samples per linear measure, e.g. samples per mm) is at least twice the highest frequency present in the image. This 'sample at twice the maximum frequency' rule is known as *Shannon's sampling theory* and the rate of sampling is called *Nyquist* frequency. (The ideas of spatial frequency are covered in Chapter Four.) Thus a digitised image obtained by sampling at the Nyquist rate contains as much information as the original.

In practice, Shannon's sampling theory is only a guide to the number of samples that have to be taken – in general signal processing applications over sampling by factors of 3 to 10 are not uncommon. The reason for this is that Shannon's sampling theory is concerned with the number of samples needed to recreate the original, not with the adequacy of the digitisation for any particular type of processing or presentation. In digital image processing, the number of samples is usually severely limited by the amount of storage available and the time taken to operate on the image. For example, a typical 512×512 image (image sizes are usually a power of 2) typically takes 128K of memory to store. (This resolution should be compared to that of a standard TV image of approximately 600×400. However, the perceived quality of a TV image is higher than these figures would suggest because the eye integrates information provided by 50 updated images per second.)

The number of samples taken is one aspect of spatial digitisation, but there is also the question of how the samples should be taken. Shannon's sampling theory assumes that the samples are *point samples* i.e. the value of f(x,y) at the sampling location, but other sampling schemes are possible. For example, you can average the value of f(x,y) over the small region (usually square) represented by the sample. Such sampling schemes can offer improvements in representation, but in practice the type of sampling used is generally determined by the type of digitising device available and this is usually some form of imperfect point sampling.

Grey level quantisation

Grey level quantisation is necessary because of the need to conserve storage and improve processing times. In principle, f(x,y) and the elements of A can take on any real value between 0 and MAX (the brightest point in the image), but in practice it is normal to restrict the brightness values to a finite set of integers. This conserves storage because only n bits are required to represent integers in the range 0 to 2^n. This means that, rather than the 16 or 32 bits needed to store a single real value, an image with integer grey levels in the range 0 to 7 needs only 3 bits per point. If 8 grey levels seem too few to represent an image, then 256 levels only need 8 bits per point, and this is still an improvement over storing a real 32 or 64 bit value for each point. A second advantage of a restricted integer range is that integer arithmetic and integer operations in general are simpler, and hence faster, than the equivalent fixed or floating point operations. Indeed, as storage requirements and processing times for each point are multiplied by the number of points in the image (for a standard serial computer at least), then it is only by the use of grey level quantisation that images can be processed in a reasonable amount of time.

We still have to describe how values of f(x,y) are assigned to an integer in the range 0 to N. The most obvious method is to divide the range 0 to MAX into N+1 equal intervals – *linear quantisation* - but it can be an advantage to use unequal intervals – *non-linear quantisation* – so that frequently occurring grey levels are covered by more integers. Using a variable or tapered quantisation can improve the average accuracy of quantisation.

There is also another reason for using non-linear quantisation which relates to the way humans perceive brightness. The eye has a logarithmic response to

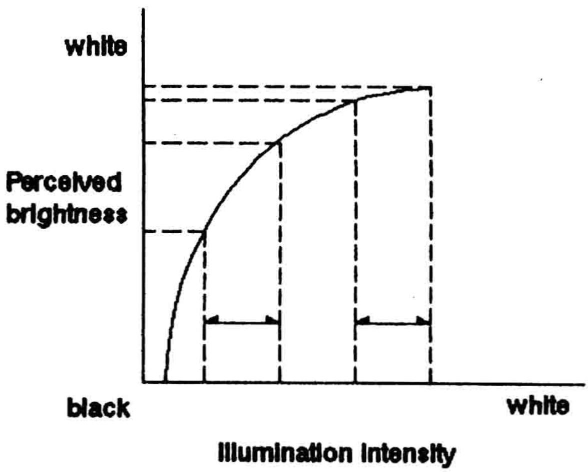


Fig. 1.3 The eye's logarithmic response to brightness

brightness that makes it more sensitive to difference in darker tones (see Fig. 1.3.)

For example, if you look at a *grey scale*, that is, a set of bars of steadily increasing brightness, with even steps in brightness between each bar, then there will appear to be a bigger difference between the darker bars. If you look at a grey scale where the steps in brightness between each bar follow a logarithmic law, then the differences will be perceived as equal. Another way of looking at this result is to say that the eye is more sensitive to changes in dark regions than in light regions of an image, and this fact is sometimes made use of in image enhancement.

Because of this logarithmic response of the eye it is usual to quantise the brightness using a logarithmic scale. This allows more bits to be used to represent tones in the darker end of the range. Even so, if too few bits are used a defect called *contouring* can often be seen in dark slowly changing areas of an image. Contouring is the appearance of false edges in an image due to the inability of the grey levels to change smoothly from one value to another.

The majority of image processing equipment available today uses 8 bits to represent its grey scale. This provides 256 grey levels and, when combined

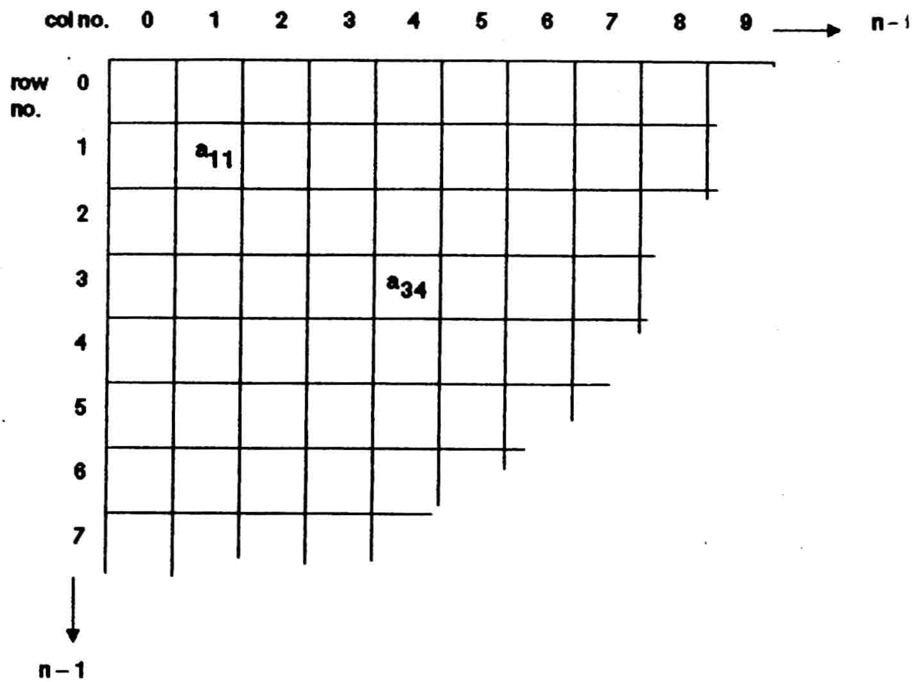


Fig. 1.4 Specifying points in an image array