Preface

Why did we write this book?

We will no doubt be asked many times: why on earth write a new book on computer vision? Fair question: there are already many good books on computer vision already out in the bookshops, as you will find referenced later, so why add to them? Part of the answer is that any textbook is a snapshot of material that exists prior to it. Computer vision, the art of processing images stored within a computer, has seen a considerable amount of research by highly qualified people and the volume of research would appear to have increased in recent years. That means a lot of new techniques have been developed, and many of the more recent approaches have yet to migrate to textbooks.

But it is not just the new research: part of the speedy advance in computer vision technique has left some areas covered only in scant detail. By the nature of research, one cannot publish material on technique that is seen more to fill historical gaps, rather than to advance knowledge. This is again where a new text can contribute.

Finally, the technology itself continues to advance. This means that there is new hardware, new programming languages and new programming environments. In particular for computer vision, the advance of technology means that computing power and memory are now relatively cheap. It is certainly considerably cheaper than when computer vision was starting as a research field. One of the authors here notes that the laptop that his portion of the book was written on has more memory, is faster, has bigger disk space and better graphics than the computer that served the entire university of his student days. And he is not that old! One of the more advantageous recent changes brought by progress has been the development of mathematical programming systems. These allow us to concentrate on mathematical technique itself, rather than on implementation detail. There are several sophisticated flavours of which Mathcad and Matlab, the chosen vehicles here, are amongst the most popular. We have been using these techniques in research and in teaching and we would argue that they have been of considerable benefit there. In research, they help us to develop technique faster and to evaluate its final implementation. For teaching, the power of a modern laptop and a mathematical system combine to show students, in lectures and in study, not only how techniques are implemented, but also how and why they work with an explicit relation to conventional teaching material.

We wrote this book for these reasons. There is a host of material we could have included but chose to omit. Our apologies to other academics if it was your own, or your favourite, technique. By virtue of the enormous breadth of the subject of computer vision, we restricted the focus to feature extraction for this has not only been the focus of our research, but it is also where the attention of established textbooks, with some exceptions, can be rather scanty. It is, however, one of the prime targets of applied computer vision, so would benefit from better attention. We have aimed to clarify some of its origins and development, whilst also exposing implementation using mathematical system. As such, we have written this text with our original aims in mind.

The book and its support

Each chapter of the book presents a particular package of information concerning feature extraction in image processing and computer vision. Each package is developed from its origins and later referenced to more recent material. Naturally, there is often theoretical development prior to implementation (in Mathcad or Matlab). We have provided working implementations of most of the major techniques we describe, and applied them to process a selection of imagery. Though the focus of our work has been more in analysing medical imagery or in biometrics (the science of recognising people by behavioural or physiological characteristic, like face recognition), the techniques are general and can migrate to other application domains.

You will find a host of further supporting information at the book's website `http://www.ecs.soton.ac.uk/~msn/book/`. First, you will find the worksheets (the Matlab and Mathcad implementations that support the text) so that you can study the techniques described herein. There are also lecturing versions that have been arranged for display via an overhead projector, with enlarged text and more interactive demonstration. The example questions (and – eventually – their answers) are also there too. The website will be kept as up to date as possible, for it also contains links to other material such as websites devoted to techniques and to applications, as well as to available software and on-line literature. Finally, any errata will be reported there. It is our regret and our responsibility that these will exist, but our inducement for their reporting concerns a pint of beer. If you find an error that we don't know about (not typos like spelling, grammar and layout) then use the mailto on the website and we shall send you a pint of good English *beer*, free!

There is a certain amount of mathematics in this book. The target audience is for third or fourth year students in BSc/BEng/MEng students in electrical or electronic engineering, or in mathematics or physics, and this is the level of mathematical analysis here. Computer vision can be thought of as a branch of applied mathematics, though this does not really apply to some areas within its remit, but certainly applies to the material herein. The mathematics essentially concerns mainly calculus and geometry though some of it is rather more detailed than the constraints of a conventional lecture course might allow. Certainly, not all the material here is covered in detail in undergraduate courses at Southampton.

The book starts with an overview of computer vision hardware, software and established material, with reference to the most sophisticated vision system yet 'developed': the *human vision* system. Though the precise details of the nature of processing that allows us to see have yet to be determined, there is a considerable range of *hardware* and *software* that allow us to give a computer system the capability to acquire, process and reason with imagery, the function of 'sight'. The first chapter also provides a comprehensive *bibliography* of material you can find on the subject, not only including textbooks, but also available software and other material. As this will no doubt be subject to change, it might well be worth consulting the website for more up-to-date information. The preference for journal references are those which are likely to be found in local university libraries, *IEEE Transactions* in particular. These are often subscribed to as they are relatively low cost, and are often of very high quality.

The next chapter concerns the basics of signal processing theory for use in computer vision. It introduces the Fourier transform that allows you to look at a signal in a new way, in terms of its frequency content. It also allows us to work out the minimum size of a picture to conserve information, to analyse the content in terms of frequency and even helps to speed up some of the later vision algorithms. Unfortunately, it does involve a few equations, but it is a new way of looking at data and at signals, and proves to be a rewarding topic of study in its own right.

We then start to look at *basic* image processing techniques, where image points are mapped into a new value first by considering a single point in an original image, and then by considering groups of points. Not only do we see common operations to make a picture's appearance better, especially for human vision, but also we see how to reduce the effects of different types of commonly encountered image noise. This is where the techniques are implemented as algorithms in Mathcad and Matlab to show precisely how the equations work.

The following chapter concerns *low-level features* which are the techniques that describe the content of an image, at the level of a whole image rather than in distinct regions of it. One of the most important processes we shall meet is called *edge detection*. Essentially, this reduces an image to a form of a caricaturist's sketch, though without a caricaturist's exaggerations. The major techniques are presented in detail, together with descriptions of their implementation. Other image properties we can derive include measures of *curvature* and measures of *movement*. These also are covered in this chapter.

These edges, the curvature or the motion need to be grouped in some way so that we can find shapes in an image. Our first approach to *shape extraction* concerns analysing the *match* of low-level information to a known template of a target shape. As this can be computationally very cumbersome, we then progress to a technique that improves computational performance, whilst maintaining an optimal performance. The technique is known as the *Hough transform* and it has long been a popular target for researchers in computer vision who have sought to clarify its basis, improve it speed, and to increase its accuracy and robustness. Essentially, by the Hough transform we estimate the parameters that govern a shape's appearance, where the shapes range from *lines* to *ellipses* and even to *unknown shapes*.

Some applications of shape extraction require to determine rather more than the parameters that control appearance, but require to be able to *deform* or *flex* to match the image template. For this reason, the chapter on shape extraction by matching is followed by one on *flexible shape* analysis. This is a topic that has shown considerable progress of late, especially with the introduction of *snakes* (*active contours*). These seek to match a shape to an image by analysing local properties. Further, we shall see how we can describe a shape by its *symmetry* and also how global constraints concerning the *statistics* of a shape's appearance can be used to guide final extraction.

Up to this point, we have not considered techniques that can be used to describe the shape found in an image. We shall find that the two major approaches concern techniques that describe a shape's perimeter and those that describe its area. Some of the *perimeter description* techniques, the Fourier descriptors, are even couched using Fourier transform theory that allows analysis of their frequency content. One of the major approaches to *area description*, statistical moments, also has a form of access to frequency components, but is of a very different nature to the Fourier analysis.

The final chapter describes *texture* analysis, prior to some introductory material on *pattern classification*. Texture describes patterns with no known analytical description and has been the target of considerable research in computer vision and image processing. It is used here more as a vehicle for the material that precedes it, such as the Fourier transform and area descriptions though references are provided for access to other generic material. There is also introductory material on how to classify these patterns against known data but again this is a window on a much larger area, to which appropriate pointers are given.

The *appendices* include material that is germane to the text, such as *co-ordinate geometry* and the method of *least squares*, aimed to be a short introduction for the reader. Other related material is referenced throughout the text, especially to on-line material. The appendices include a printout of one of the shortest of the Mathcad and Matlab *worksheets*.

In this way, the text covers all major areas of feature extraction in image processing and computer vision. There is considerably more material in the subject than is presented here: for example, there is an enormous volume of material in 3D computer vision and in 2D signal processing which is only alluded to here. But to include all that would lead to a monstrous book that no one could afford, or even pick up! So we admit we give a snapshot, but hope more that it is considered to open another window on a fascinating and rewarding subject.

In gratitude

We are immensely grateful to the input of our colleagues, in particular to Dr Steve Gunn and to Dr John Carter. The family who put up with it are Maria Eugenia and Caz and the nippers. We are also very grateful to past and present researchers in computer vision at the Image, Speech and Intelligent Systems Research Group (formerly the Vision, Speech and Signal Processing Group) under (or who have survived?) Mark's supervision at the Department of Electronics and Computer Science, University of Southampton. These include: Dr Hani Muammar, Dr Xiaoguang Jia, Dr Yan Chen, Dr Adrian Evans, Dr Colin Davies, Dr David Cunado, Dr Jason Nash, Dr Ping Huang, Dr Liang Ng, Dr Hugh Lewis, David Benn, Douglas Bradshaw, David Hurley, Mike Grant, Bob Roddis, Karl Sharman, Jamie Shutler, Jun Chen, Andy Tatem, Chew Yam, James Hayfron-Acquah, Yalin Zheng and Jeff Foster. We are also very grateful to past Southampton students on BEng and MEng Electronic Engineering, MEng Information Engineering, BEng and MEng Computer Engineering and BSc Computer Science who have pointed our earlier mistakes, noted areas for clarification and in some cases volunteered some of the material herein. To all of you, our very grateful thanks.

Final message

We ourselves have already benefited much by writing this book. As we already know, previous students have also benefited, and contributed to it as well. But it remains our hope that it does inspire people to join in this fascinating and rewarding subject that has proved to be such a source of pleasure and inspiration to its many workers.

Mark S. Nixon
University of Southampton

Alberto S. Aguado
University of Surrey