

---

John A. Richards · Xiuping Jia

Remote Sensing Digital Image Analysis

John A. Richards · Xiuping Jia

# **Remote Sensing Digital Image Analysis**

An Introduction

4th Edition

With 197 Figures

John A. Richards  
The Australian National University  
Research School of Information Sciences  
and Engineering (RSISE)  
Dept. of Systems Engineering  
Canberra, ACT 0200  
Australia  
*John.Richards@anu.edu.au*

Xiuping Jia  
The University of New South Wales  
Australian Defence Force Academy  
School of Information Technology and Electrical Engineering  
Northcott Drive Cambell ACT 2600  
Australia  
*x-jia@adfa.edu.au*

*Front Cover*

*Image of the city of Canberra, the national capital of Australia, recorded by the HyMap scanner manufactured by Integrated Spectronics Pty Ltd, Sydney, Australia*

Library of Congress Control Number: 2005926341

ISBN-10 3-540-25128-6 Springer Berlin Heidelberg New York  
ISBN-13 978-3-540-25128-6 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in other ways, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under German Copyright Law.

**Springer is a part of Springer Science+Business Media**  
[springeronline.com](http://springeronline.com)

© Springer-Verlag Berlin Heidelberg 2006  
Printed in Germany

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: PTP-Berlin Protago-T<sub>E</sub>X-Production GmbH, Germany  
Final processing by PTP-Berlin Protago-T<sub>E</sub>X-Production GmbH, Germany  
Cover-Design: Struve & Partner, Heidelberg  
Printed on acid-free paper 62/3141/Yu - 5 4 3 2 1 0

## Preface to the Fourth Edition

This fourth edition has been developed to reflect the changes that have occurred in techniques for the analysis of digital image data in remote sensing over the past five years or so. Its focus is on those procedures that seem now to have become part of the set of tools regularly used to perform thematic mapping. As with previous revisions, the fundamental material has been preserved in its original form because of its tutorial value; its style has been revised in places and it has been supplemented if newer aspects have emerged in the time since the third edition appeared. The theme of the book remains, however, on the needs of the senior student and practitioner.

The earlier editions have contained extensive material in Chapter 1 on satellite programs and sensor characteristics. Although that material is important in the context of understanding the application of image analysis procedures, the rapid development of quasi- and fully operational programs in the past decade has meant that expanding the material of Chap. 1, as required, would have distracted from the role of that chapter to introduce the reader to the nature and properties of digital image data in remote sensing. Accordingly, all of the material on satellite programs and sensor specifications has been moved to a new Appendix A. Chapter 1 has then been completely rewritten as a stand-alone introduction to sensors in general and the data properties of importance in image analysis.

Many changes have been made throughout the book to meet the increasing emphasis on hyperspectral data and its analysis. Although much of that is contained in Chap. 13, techniques required for hyperspectral data processing are developed in the context of previous chapters, particularly new material on feature extraction tools that work well on hyperspectral data sets; they are covered in Chap. 10.

Chapter 10 has been re-named. It was felt that there is too much confusion in the term Data Fusion to retain it as the title for material that is fundamentally concerned with thematic mapping from multiple data sources and multiple sensors.

Chapter 8, dealing with supervised classification methods, has been substantially supplemented. Sections have been incorporated on  $k$  nearest neighbour classification, Markov random fields and support vector classifiers.

Other modifications relate to the noise adjusted principal components transformation, the definition of texture, a re-working of the contrast modification material and the inclusion of several further illustrations and problems.

The authors continue to enjoy the very strong support and understanding of their families, so important in the undertaking of this work, for which they express and record their sincere gratitude.

Canberra, Australia, January 2005

John A. Richards  
Xiuping Jia

# Preface to the Third Edition

In the time since the second edition of this text was produced two significant trends have been apparent. First, access to image processing technology has continued to improve significantly with most students and practitioners now having readily available inexpensive workstations and powerful software for analysing and manipulating image data.

The second change has been the dramatic increase in the numbers of satellite, aircraft and sensor programs. Perhaps most significant is the widespread availability of hyperspectral data and the special challenges presented by that data type for information extraction.

Accordingly, this third edition has been written to reflect those trends while, at the same time, preserving the important elements of image processing and analysis algorithms of significance in remote sensing applications.

The major changes between the previous edition and this are an update of Chap. 1, and the introduction of a new Chap. 13 dealing with methods for analysing hyperspectral data sets.

Chapter 12 has also been significantly altered to provide a focus on the interpretation of data sets that are mixed and could contain, for example, different types of imagery along with other spatial data types found in a geographic information system. The previous knowledge-based material has been retained but material originally covered in Chap. 8 dealing with multi-source data analysis has been combined with knowledge-based methods to create this chapter on data fusion.

The authors wish to express their appreciation to their colleagues for the assistance they have received in preparing this new edition. In particular David Landgrebe of Purdue University remains a great supporter and both authors have had the benefit of working with Dave over the years, including periods spent at Purdue.

The authors are also enormously grateful to their families for their understanding and support in making the completion of this third edition possible.

Canberra, Australia, August 1998

John A. Richards  
Xiuping Jia

## Preface to the Second Edition

Possibly the greatest change confronting the practitioner and student of remote sensing in the period since the first edition of this text appeared in 1986 has been the enormous improvement in accessibility to image processing technology. Falling hardware and software costs, combined with an increase in functionality through the development of extremely versatile user interfaces, has meant that even the user unskilled in computing now has immediate and ready access to powerful and flexible means for digital image analysis and enhancement. An understanding, at algorithmic level, of the various methods for image processing has become therefore even more important in the past few years to ensure the full capability of digital image processing is utilised.

This period has also been a busy one in relation to digital data supply. Several nations have become satellite data gatherers and providers, using both optical and microwave technology. Practitioners and researchers are now faced, therefore, with the need to be able to process imagery from several sensors, together with other forms of spatial data. This has been driven, to an extent, by developments in Geographic Information Systems (GIS) which, in turn, have led to the appearance of newer image processing procedures as adjuncts to more traditional approaches.

The additional material incorporated in this edition addresses these changes. First, Chap. 1 has been significantly revised to reflect developments in satellite and sensor programs. Removal of information on older systems has been resisted since their data is part of an important archive which still finds value, particularly for historical applications.

Chapter 8, dealing with supervised classification methods, has been substantially increased to allow context classification to be addressed, along with techniques, such as evidential processing and neural networks, which show promise as viable image interpretation tools. While the inclusion of these topics has caused the chapter to become particularly large in comparison to the others, it was felt important not to separate the material from the more traditional methods. The chapter is now presented therefore in two parts – the first covers the standard supervised classification procedures and the second the new topics.

A departure from the application of classical digital image processing to remote sensing over the last five years has been the adoption of knowledge-based methods, where qualitative rather than quantitative reasoning is used to perform interpretations. This is the subject of a new chapter which seeks to introduce reasoning based on knowledge as a means for single image interpretation, and as an approach that can deal successfully with the mixed spatial data types of a GIS.

Besides these changes, the opportunity has been taken to correct typographical and related errors in the first edition and to bring other material up to date.

As with the first edition, the author wishes to record his appreciation to others for their support and assistance. Ashwin Srinivasan, now with the Turing Institute in Glasgow, as a very gifted graduate student, developed much of the material on which Chap. 12 is based and, during his time as a student, helped the author understand the mechanisms of knowledge processing. He also kindly read and offered comments on that chapter. The author's colleague Don Fraser similarly read and provided comments on the material on neural networks. Philip Swain and David Landgrebe of Purdue University continue to support the author in many ways: through their feedback on the first edition, their interaction on image processing, and Dave's provision of MultiSpec, the Macintosh computer version of the LARSSYS software package. Finally, the author again expresses gratitude to his family for their constant support, without which the energy and enthusiasm needed to complete this edition might not have been found.

Canberra, Australia, March 1993

John A. Richards

## Preface to the First Edition

With the widespread availability of satellite and aircraft remote sensing image data in digital form, and the ready access most remote sensing practitioners have to computing systems for image interpretation, there is a need to draw together the range of digital image processing procedures and methodologies commonly used in this field into a single treatment. It is the intention of this book to provide such a function, at a level meaningful to the non-specialist digital image analyst, but in sufficient detail that algorithm limitations, alternative procedures and current trends can be appreciated. Often the applications specialist in remote sensing wishing to make use of digital processing procedures has had to depend upon either the mathematically detailed treatments of image processing found in the electrical engineering and computer science literature, or the sometimes necessarily superficial treatments given in general texts on remote sensing. This book seeks to redress that situation.

Both image enhancement and classification techniques are covered making the material relevant in those applications in which photointerpretation is used for information extraction and in those wherein information is obtained by classification. It grew out of a graduate course on digital image processing and analysis techniques for remote sensing data given annually since 1980 at the University of New South Wales. If used as a graduate textbook its contents with the exception of Chap. 7 can be covered substantially in a single semester. Its function as a text is supported by the provision of exercises at the end of each chapter. Most do not require access to a computer for solution. Rather they are capable of hand manipulation and are included to highlight important issues. In many cases some new material is introduced by means of these exercises.

Each chapter concludes with a short critical bibliography that points to more detailed treatments of specific topics and provides, where appropriate, comment on techniques of marginal interest to the mainstream of the book's theme.

Chapter 1 is essentially a compendium of data sources commonly encountered in digital form in remote sensing. It is provided as supporting material for the chapters that follow, drawing out the particular properties of each data source of importance. The second chapter deals with radiometric and geometric errors in image data and

with means for correction. This also contains material on registration of images to maps and images to each other. Here, as in all techniques chapters, real and modelled image data examples are given. Chapter 3 establishes the role of computer processing both for photointerpretation by a human analyst and for machine analysis. This may be skipped by the remote sensing professional but is an important position chapter if the book is to be used in teaching.

Chapters 4 and 5 respectively cover the range of radiometric and geometric enhancement techniques commonly adopted in practice, while Chap. 6 is addressed to multispectral transformations of data. This includes the principal components transformation and image arithmetic. Chapter 7 is given over to Fourier transformations. This material is becoming more important in remote sensing with falling hardware costs and the ready availability of peripheral array processors. Here the properties of discrete Fourier analysis are given along with means by which the fast Fourier transform algorithm can be used on image data.

Chapters 8, 9 and 10 provide a treatment of the tools used in image classification, commencing with supervised classification methods, moving through commonly used clustering algorithms for unsupervised classification and concluding with means for separability analysis. These are drawn together into classification methodologies in Chap. 11 which also provides a set of case studies.

Even though the treatment provided is intended for the non-specialist image analyst, it is still necessary that it be cast in the context of some vector and matrix algebra. Otherwise it would be impracticable. Consequently, an appendix is provided on essential results on vectors and matrices, and all important points in the text are illustrated by simple worked examples. These demonstrate how vector operations are evaluated. Beyond this material it is assumed the reader has a passing knowledge of basic probability and statistics including an appreciation of the multivariate normal distribution.

Several other appendices are provided to supplement the main presentation. One deals with developments in image processing hardware and particularly the architecture (in block form) of interactive image display sub-systems. This material highlights trends towards hardware implementation of image processing and illustrates how many of the algorithms presented in the book can be executed in near real time.

Owing to common practice, some decisions have had to be taken in relation to definitions even though they could offend the purist. For example the term "pixel" strictly refers to a unit of digital image data and not to an area on the ground. The latter is more properly called an effective ground resolution element. However because the practice of referring to ground resolution elements as pixels, dimensioned in metres, is so widespread, the current treatment seeks not to be pedantic but rather follows common practice for simplicity. A difficulty also arises with respect to the numbering chosen for the wavebands in the Landsat multispectral scanner. Historically these have been referred to as bands 4 to 7 for Landsats 1 to 3. From Landsat 4 onwards they have been renumbered as bands 1 to 4. The convention adopted herein is mixed. When a particular satellite is evident in the discussion, the respective convention is

adopted and is clear from the context of that discussion. In other cases the convention for Landsat 4 has been used as much as possible.

Finally, it is a pleasure to acknowledge the contributions made by others to the production of this book. The manuscript was typed by Mrs Moo Song and Mrs Alisa Moen, both of whom undertook the task tirelessly and with great patience and forbearance. Assistance with computing was given by Leanne Bischof, at all times cheerfully and accurately. The author's colleagues and students also played their part, both through direct discussion and by that process of gradual learning that occurs over many years of association. Particular thanks are expressed to two people. The author counts himself fortunate to be a friend and colleague of Professor Philip Swain of Purdue University, who in his own way, has had quite an impact on the author's thinking about digital data analysis, particularly in remote sensing. Also, the author has had the good fortune to work with Tong Lee, a graduate student with extraordinary insight and ability, who also has contributed to the material through his many discussions with the author on the theoretical foundations of digital image processing.

The support and encouragement the author has received from his family during the preparation of this work has been immeasurable. It is fitting therefore to conclude in gratitude to Glenda, Matthew and Jennifer, for their understanding and enthusiasm.

Kensington, Australia, May 1986

John A. Richards

# Contents

<b>1</b>	<b>Sources and Characteristics of Remote Sensing Image Data . . . . .</b>	<b>1</b>
1.1	Introduction to Data Sources . . . . .	1
1.1.1	Characteristics of Digital Image Data . . . . .	1
1.1.2	Spectral Ranges Commonly Used in Remote Sensing . . . . .	4
1.1.3	Concluding Remarks . . . . .	8
1.2	Remote Sensing Platforms . . . . .	9
1.3	Image Data Sources in the Microwave Region . . . . .	12
1.3.1	Side Looking Airborne Radar and Synthetic Aperture Radar . . . . .	12
1.4	Spatial Data Sources in General . . . . .	15
1.4.1	Types of Spatial Data . . . . .	15
1.4.2	Data Formats . . . . .	17
1.4.3	Geographic Information Systems (GIS) . . . . .	18
1.4.4	The Challenge to Image Processing and Analysis . . . . .	20
1.5	A Comparison of Scales in Digital Image Data . . . . .	21
<b>2</b>	<b>Error Correction and Registration of Image Data . . . . .</b>	<b>27</b>
2.1	Sources of Radiometric Distortion . . . . .	27
2.1.1	The Effect of the Atmosphere on Radiation . . . . .	28
2.1.2	Atmospheric Effects on Remote Sensing Imagery . . . . .	31
2.1.3	Instrumentation Errors . . . . .	31
2.2	Correction of Radiometric Distortion . . . . .	32
2.2.1	Detailed Correction of Atmospheric Effects . . . . .	33
2.2.2	Bulk Correction of Atmospheric Effects . . . . .	34
2.2.3	Correction of Instrumentation Errors . . . . .	36
2.3	Sources of Geometric Distortion . . . . .	37
2.3.1	Earth Rotation Effects . . . . .	38
2.3.2	Panoramic Distortion . . . . .	39
2.3.3	Earth Curvature . . . . .	42
2.3.4	Scan Time Skew . . . . .	43

2.3.5	Variations in Platform Altitude, Velocity and Attitude .....	43
2.3.6	Aspect Ratio Distortion .....	44
2.3.7	Sensor Scan Nonlinearities .....	45
2.4	Correction of Geometric Distortion .....	46
2.4.1	Use of Mapping Polynomials for Image Correction .....	46
2.4.1.1	Mapping Polynomials and Ground Control Points .....	47
2.4.1.2	Resampling .....	48
2.4.1.3	Interpolation .....	48
2.4.1.4	Choice of Control Points .....	51
2.4.1.5	Example of Registration to a Map Grid .....	51
2.4.2	Mathematical Modelling .....	54
2.4.2.1	Aspect Ratio Correction .....	54
2.4.2.2	Earth Rotation Skew Correction .....	55
2.4.2.3	Image Orientation to North-South .....	55
2.4.2.4	Correction of Panoramic Effects .....	55
2.4.2.5	Combining the Corrections .....	56
2.5	Image Registration .....	56
2.5.1	Georeferencing and Geocoding .....	56
2.5.2	Image to Image Registration .....	57
2.5.3	Control Point Localisation by Correlation .....	57
2.5.4	Example of Image to Image Registration .....	58
2.6	Miscellaneous Image Geometry Operations .....	59
2.6.1	Image Rotation .....	61
2.6.2	Scale Changing and Zooming .....	61
<b>3</b>	<b>The Interpretation of Digital Image Data</b> .....	67
3.1	Approaches to Interpretation .....	67
3.2	Forms of Imagery for Photointerpretation .....	69
3.3	Computer Processing for Photointerpretation .....	72
3.4	An Introduction to Quantitative Analysis – Classification .....	72
3.5	Multispectral Space and Spectral Classes .....	75
3.6	Quantitative Analysis by Pattern Recognition .....	77
3.6.1	Pixel Vectors and Labelling .....	77
3.6.2	Unsupervised Classification .....	78
3.6.3	Supervised Classification .....	78
<b>4</b>	<b>Radiometric Enhancement Techniques</b> .....	83
4.1	Introduction .....	83
4.1.1	Point Operations and Look Up Tables .....	83
4.1.2	Scalar and Vector Images .....	83
4.2	The Image Histogram .....	84
4.3	Contrast Modification in Image Data .....	84
4.3.1	Histogram Modification Rule .....	84
4.3.2	Linear Contrast Modification .....	86

4.3.3	Saturating Linear Contrast Enhancement . . . . .	88
4.3.4	Automatic Contrast Enhancement . . . . .	88
4.3.5	Logarithmic and Exponential Contrast Enhancement . . . . .	89
4.3.6	Piecewise Linear Contrast Modification . . . . .	89
4.4	Histogram Equalization . . . . .	90
4.4.1	Use of the Cumulative Histogram . . . . .	90
4.4.2	Anomalies in Histogram Equalization . . . . .	95
4.5	Histogram Matching . . . . .	97
4.5.1	Principle of Histogram Matching . . . . .	97
4.5.2	Image to Image Contrast Matching . . . . .	98
4.5.3	Matching to a Mathematical Reference . . . . .	99
4.6	Density Slicing . . . . .	101
4.6.1	Black and White Density Slicing . . . . .	101
4.6.2	Colour Density Slicing and Pseudocolouring . . . . .	104
<b>5</b>	<b>Geometric Enhancement Using Image Domain Techniques</b> . . . . .	109
5.1	Neighbourhood Operations . . . . .	109
5.2	Template Operators . . . . .	109
5.3	Geometric Enhancement as a Convolution Operation . . . . .	110
5.4	Image Domain Versus Fourier Transformation Approaches . . . . .	113
5.5	Image Smoothing (Low Pass Filtering) . . . . .	115
5.5.1	Mean Value Smoothing . . . . .	115
5.5.2	Median Filtering . . . . .	116
5.6	Edge Detection and Enhancement . . . . .	118
5.6.1	Linear Edge Detecting Templates . . . . .	120
5.6.2	Spatial Derivative Techniques . . . . .	121
5.6.2.1	The Roberts Operator . . . . .	121
5.6.2.2	The Sobel Operator . . . . .	122
5.6.2.3	The Prewitt Operator . . . . .	122
5.6.3	Thinning, Linking and Border Responses . . . . .	123
5.6.4	Edge Enhancement by Subtractive Smoothing (Sharpening) . . . . .	123
5.7	Line Detection . . . . .	125
5.7.1	Linear Line Detecting Templates . . . . .	125
5.7.2	Non-linear and Semi-linear Line Detecting Templates . . . . .	125
5.8	General Convolution Filtering . . . . .	127
5.9	Detecting Geometric Properties . . . . .	128
5.9.1	Texture . . . . .	128
5.9.2	Spatial Correlation – The Semivariogram . . . . .	131
5.9.3	Shape Detection . . . . .	132
<b>6</b>	<b>Multispectral Transformations of Image Data</b> . . . . .	137
6.1	The Principal Components Transformation . . . . .	137
6.1.1	The Mean Vector and Covariance Matrix . . . . .	138
6.1.2	A Zero Correlation, Rotational Transform . . . . .	141

6.1.3	Examples – Some Practical Considerations .....	145
6.1.4	The Effect of an Origin Shift .....	150
6.1.5	Application of Principal Components in Image Enhancement and Display .....	150
6.1.6	The Taylor Method of Contrast Enhancement .....	151
6.1.7	Other Applications of Principal Components Analysis .....	154
6.2	Noise Adjusted Principal Components Transformation .....	154
6.3	The Kauth-Thomas Tasseled Cap Transformation .....	156
6.4	Image Arithmetic, Band Ratios and Vegetation Indices .....	160
<b>7</b>	<b>Fourier Transformation of Image Data .....</b>	<b>165</b>
7.1	Introduction .....	165
7.2	Special Functions .....	165
7.2.1	The Complex Exponential Function .....	166
7.2.2	The Dirac Delta Function .....	166
7.2.2.1	Properties of the Delta Function .....	167
7.2.3	The Heaviside Step Function .....	168
7.3	Fourier Series .....	168
7.4	The Fourier Transform .....	169
7.5	Convolution .....	171
7.5.1	The Convolution Integral .....	171
7.5.2	Convolution with an Impulse .....	171
7.5.3	The Convolution Theorem .....	173
7.6	Sampling Theory .....	173
7.7	The Discrete Fourier Transform .....	176
7.7.1	The Discrete Spectrum .....	176
7.7.2	Discrete Fourier Transform Formulae .....	177
7.7.3	Properties of the Discrete Fourier Transform .....	178
7.7.4	Computation of the Discrete Fourier Transform .....	179
7.7.5	Development of the Fast Fourier Transform Algorithm .....	179
7.7.6	Computational Cost of the Fast Fourier Transform .....	183
7.7.7	Bit Shuffling and Storage Considerations .....	184
7.8	The Discrete Fourier Transform of an Image .....	184
7.8.1	Definition .....	184
7.8.2	Evaluation of the Two Dimensional, Discrete Fourier Transform .....	185
7.8.3	The Concept of Spatial Frequency .....	185
7.8.4	Image Filtering for Geometric Enhancement .....	187
7.8.5	Convolution in Two Dimensions .....	188
7.9	Concluding Remarks .....	189
<b>8</b>	<b>Supervised Classification Techniques .....</b>	<b>193</b>
8.1	Steps in Supervised Classification .....	193
8.2	Maximum Likelihood Classification .....	194
8.2.1	Bayes' Classification .....	194

8.2.2	The Maximum Likelihood Decision Rule . . . . .	195
8.2.3	Multivariate Normal Class Models . . . . .	196
8.2.4	Decision Surfaces . . . . .	196
8.2.5	Thresholds . . . . .	197
8.2.6	Number of Training Pixels Required for Each Class . . . . .	199
8.2.7	A Simple Illustration . . . . .	199
8.3	Minimum Distance Classification . . . . .	201
8.3.1	The Case of Limited Training Data . . . . .	201
8.3.2	The Discriminant Function . . . . .	202
8.3.3	Degeneration of Maximum Likelihood to Minimum Distance Classification . . . . .	203
8.3.4	Decision Surfaces . . . . .	204
8.3.5	Thresholds . . . . .	204
8.4	Parallelepiped Classification . . . . .	204
8.5	Classification Time Comparison of the Classifiers . . . . .	206
8.6	Other Supervised Approaches . . . . .	206
8.6.1	The Mahalanobis Classifier . . . . .	206
8.6.2	Table Look Up Classification . . . . .	207
8.6.3	The $kNN$ (Nearest Neighbour) Classifier . . . . .	207
8.7	Gaussian Mixture Models . . . . .	208
8.8	Context Classification . . . . .	209
8.8.1	The Concept of Spatial Context . . . . .	209
8.8.2	Context Classification by Image Pre-processing . . . . .	210
8.8.3	Post Classification Filtering . . . . .	211
8.8.4	Probabilistic Label Relaxation . . . . .	211
8.8.4.1	The Basic Algorithm . . . . .	211
8.8.4.2	The Neighbourhood Function . . . . .	212
8.8.4.3	Determining the Compatibility Coefficients . . . . .	213
8.8.4.4	The Final Step – Stopping the Process . . . . .	214
8.8.4.5	Examples . . . . .	215
8.8.5	Handling Spatial Context by Markov Random Fields . . . . .	216
8.9	Non-parametric Classification: Geometric Approaches . . . . .	219
8.9.1	Linear Discrimination . . . . .	220
8.9.1.1	Concept of a Weight Vector . . . . .	220
8.9.1.2	Testing Class Membership . . . . .	221
8.9.1.3	Training . . . . .	221
8.9.1.4	Setting the Correction Increment . . . . .	223
8.9.1.5	Classification – The Threshold Logic Unit . . . . .	224
8.9.1.6	Multicategory Classification . . . . .	225
8.9.2	Support Vector Classifiers . . . . .	226
8.9.2.1	Linearly Separable Data . . . . .	226
8.9.2.2	Linear Inseparability – The Use of Kernel Functions . . . . .	230
8.9.2.3	Multicategory Classification . . . . .	231

8.9.3	Networks of Classifiers – Solutions of Nonlinear Problems	231
8.9.4	The Neural Network Approach .....	232
8.9.4.1	The Processing Element .....	232
8.9.4.2	Training the Neural Network – Backpropagation .....	234
8.9.4.3	Choosing the Network Parameters .....	238
8.9.4.4	Examples .....	238
<b>9</b>	<b>Clustering and Unsupervised Classification</b> .....	249
9.1	Delineation of Spectral Classes .....	249
9.2	Similarity Metrics and Clustering Criteria .....	249
9.3	The Iterative Optimization (Migrating Means) Clustering Algorithm .....	251
9.3.1	The Basic Algorithm .....	252
9.3.2	Mergings and Deletions .....	252
9.3.3	Splitting Elongated Clusters .....	254
9.3.4	Choice of Initial Cluster Centres .....	254
9.3.5	Clustering Cost .....	254
9.4	Unsupervised Classification and Cluster Maps .....	255
9.5	A Clustering Example .....	255
9.6	A Single Pass Clustering Technique .....	257
9.6.1	Single Pass Algorithm .....	257
9.6.2	Advantages and Limitations .....	259
9.6.3	Strip Generation Parameter .....	259
9.6.4	Variations on the Single Pass Algorithm .....	259
9.6.5	An Example .....	260
9.7	Agglomerative Hierarchical Clustering .....	260
9.8	Clustering by Histogram Peak Selection .....	263
<b>10</b>	<b>Feature Reduction</b> .....	267
10.1	Feature Reduction and Separability .....	267
10.2	Separability Measures for Multivariate Normal Spectral Class Models .....	268
10.2.1	Distribution Overlaps .....	268
10.2.2	Divergence .....	269
10.2.2.1	A General Expression .....	269
10.2.2.2	Divergence of a Pair of Normal Distributions ..	270
10.2.2.3	Use of Divergence for Feature Selection .....	271
10.2.2.4	A Problem with Divergence .....	272
10.2.3	The Jeffries-Matusita (JM) Distance .....	273
10.2.3.1	Definition .....	273
10.2.3.2	Comparison of Divergence and JM Distance ..	274
10.2.4	Transformed Divergence .....	274
10.2.4.1	Definition .....	274

10.2.4.2 Relation Between Transformed Divergence and Probability of Correct Classification .....	275
10.2.4.3 Use of Transformed Divergence in Clustering .....	276
10.3 Separability Measures for Minimum Distance Classification .....	276
10.4 Feature Reduction by Data Transformation .....	276
10.4.1 Feature Reduction Using the Principal Components Transformation .....	277
10.4.2 Canonical Analysis as a Feature Selection Procedure .....	279
10.4.2.1 Within Class and Among Class Covariance Matrices .....	280
10.4.2.2 A Separability Measure .....	281
10.4.2.3 The Generalised Eigenvalue Equation .....	281
10.4.2.4 An Example .....	283
10.4.3 Discriminant Analysis Feature Extraction (DAFE) .....	285
10.4.4 Non-parametric Discriminant Analysis and Decision Boundary Feature Extraction (DBFE) .....	286
10.4.5 Non-parametric Weighted Feature Extraction (NWFE) .....	290
10.4.6 Arithmetic Transformations .....	292
<b>11 Image Classification Methodologies .....</b>	<b>295</b>
11.1 Introduction .....	295
11.2 Supervised Classification .....	295
11.2.1 Outline .....	295
11.2.2 Determination of Training Data .....	296
11.2.3 Feature Selection .....	297
11.2.4 Detecting Multimodal Distributions .....	297
11.2.5 Presentation of Results .....	298
11.2.6 Effect of Resampling on Classification .....	298
11.3 Unsupervised Classification .....	299
11.3.1 Outline, and Comparison with Supervised Methods .....	299
11.3.2 Feature Selection .....	301
11.4 A Hybrid Supervised/Unsupervised Methodology .....	301
11.4.1 The Essential Steps .....	301
11.4.2 Choice of the Clustering Regions .....	302
11.4.3 Rationalisation of the Number of Spectral Classes .....	302
11.5 Assessment of Classification Accuracy .....	303
11.5.1 Using a Testing Set of Pixels .....	303
11.5.2 The Leave One Out Method of Accuracy Assessment – Cross Validation .....	307
11.6 Case Study 1: Irrigated Area Determination .....	307
11.6.1 Background .....	308
11.6.2 The Study Region .....	308
11.6.3 Clustering .....	309
11.6.4 Signature Generation .....	312
11.6.5 Classification and Results .....	312

11.6.6	Concluding Remarks .....	312
11.7	Case Study 2: Multitemporal Monitoring of Bush Fires .....	314
11.7.1	Background .....	314
11.7.2	Simple Illustration of the Technique .....	314
11.7.3	The Study Area .....	316
11.7.4	Registration .....	316
11.7.5	Principal Components Transformation .....	317
11.7.6	Classification of Principal Components Imagery .....	319
11.8	Hierarchical Classification .....	321
11.8.1	The Decision Tree Classifier .....	321
11.8.2	Decision Tree Design .....	323
11.8.3	Progressive Two-Class Decision Classifier .....	324
11.8.4	Error Accumulation in a Decision Tree .....	327
11.9	A Note on Hyperspectral Data Classification .....	328
<b>12</b>	<b>Multisource, Multisensor Methods .....</b>	<b>333</b>
12.1	The Stacked Vector Approach .....	334
12.2	Statistical Multisource Methods .....	334
12.2.1	Joint Statistical Decision Rules .....	334
12.2.2	Committee Classifiers .....	335
12.2.3	Opinion Pools and Consensus Theoretic Methods .....	336
12.2.4	Use of Prior Probability .....	337
12.2.5	Supervised Label Relaxation .....	337
12.3	The Theory of Evidence .....	338
12.3.1	The Concept of Evidential Mass .....	338
12.3.2	Combining Evidence – the Orthogonal Sum .....	340
12.3.3	Decision Rule .....	341
12.4	Knowledge-Based Image Analysis .....	342
12.4.1	Knowledge Processing: Emulating Photointerpretation ..	342
12.4.2	Fundamentals of a Knowledge-Based Image Analysis System .....	344
12.4.2.1	Structure .....	344
12.4.2.2	Representation of Knowledge: Rules .....	345
12.4.2.3	The Inference Mechanism .....	346
12.4.3	Handling Multisource and Multisensor Data .....	347
12.4.4	An Example .....	349
12.4.4.1	Rules as Justifiers for a Labelling Proposition ..	350
12.4.4.2	Endorsement of a Labelling Proposition .....	351
12.4.4.3	Knowledge Base and Results .....	352
<b>13</b>	<b>Interpretation of Hyperspectral Image Data .....</b>	<b>359</b>
13.1	Data Characteristics .....	359
13.2	The Challenge to Interpretation .....	361
13.2.1	Data Volume .....	362
13.2.2	Redundancy .....	362

13.2.3	The Need for Calibration .....	364
13.2.4	The Problem of Dimensionality: The Hughes Phenomenon .....	364
13.3	Data Calibration Techniques .....	366
13.3.1	Detailed Radiometric Correction .....	366
13.3.2	Data Normalisation .....	367
13.3.3	Approximate Radiometric Correction .....	368
13.4	Interpretation Using Spectral Information .....	368
13.4.1	Spectral Angle Mapping .....	368
13.4.2	Using Expert Spectral Knowledge and Library Searching .....	369
13.4.3	Library Searching by Spectral Coding .....	371
13.4.3.1	Binary Spectral Codes .....	371
13.4.3.2	Matching Algorithms .....	373
13.5	Hyperspectral Interpretation by Statistical Methods .....	373
13.5.1	Limitations of Traditional Thematic Mapping Procedures .....	373
13.5.2	Block-based Maximum Likelihood Classification .....	375
13.6	Feature Reduction .....	377
13.6.1	Feature Selection .....	378
13.6.2	Spectral Transformations .....	379
13.6.3	Feature Selection from Principal Components Transformed Data .....	381
13.7	Regularised Covariance Estimators .....	381
13.8	Compression of Hyperspectral Data .....	382
13.9	Spectral Unmixing: End Member Analysis .....	385
<b>A</b>	<b>Missions and Sensors .....</b>	<b>389</b>
A.1	Weather Satellite Sensors .....	389
A.1.1	Polar Orbiting and Geosynchronous Satellites .....	389
A.1.2	The NOAA AVHRR (Advanced Very High Resolution Radiometer) .....	390
A.1.3	The Nimbus CZCS (Coastal Zone Colour Scanner) .....	390
A.1.4	GMS VISSR (Visible and Infrared Spin Scan Radiometer) and GOES Imager .....	391
A.2	Earth Resource Satellite Sensors in the Visible and Infrared Regions .....	391
A.2.1	The Landsat System .....	391
A.2.2	The Landsat Instrument Complement .....	393
A.2.3	The Return Beam Vidicon (RBV) .....	393
A.2.4	The Multispectral Scanner (MSS) .....	394
A.2.5	The Thematic Mapper (TM) and Enhanced Thematic Mapper + (ETM+) .....	396
A.2.6	The SPOT HRV, HRVIR, HRG, HRS and Vegetation Instruments .....	397

A.2.7	ADEOS (Advanced Earth Observing Satellite) . . . . .	398
A.2.8	Sea-Viewing Wide Field of View Sensor (SeaWiFS) . . . . .	399
A.2.9	Marine Observation Satellite (MOS) . . . . .	400
A.2.10	Indian Remote Sensing Satellite (IRS) . . . . .	401
A.2.11	RESURS-O1 . . . . .	401
A.2.12	The Earth Observing 1 (EO-1) Mission . . . . .	401
A.2.13	Aqua and Terra . . . . .	401
A.2.14	Ikonos . . . . .	405
A.3	Aircraft Scanners in the Visible and Infrared Regions . . . . .	405
A.3.1	General Considerations . . . . .	405
A.3.2	Airborne Imaging Spectrometers . . . . .	406
A.4	Spaceborne Imaging Radar Systems . . . . .	407
A.4.1	The Seasat SAR . . . . .	407
A.4.2	Spaceborne (Shuttle) Imaging Radar-A (SIR-A) . . . . .	407
A.4.3	Spaceborne (Shuttle) Imaging Radar-B (SIR-B) . . . . .	409
A.4.4	Spaceborne (Shuttle) Imaging Radar-C (SIR-C)/X-Band Synthetic Aperture Radar (X-SAR) . . . . .	409
A.4.5	ERS-1,2 . . . . .	409
A.4.6	JERS-1 . . . . .	410
A.4.7	Radarsat . . . . .	410
A.4.8	Shuttle Radar Topography Mission (SRTM) . . . . .	410
A.4.9	Envisat Advanced Synthetic Aperture Radar (ASAR) . . . . .	411
A.4.10	The Advanced Land Observing Satellite (ALOS) PALSAR . . . . .	411
A.5	Aircraft Imaging Radar Systems . . . . .	411
<b>B</b>	<b>Satellite Altitudes and Periods</b> . . . . .	413
<b>C</b>	<b>Binary Representation of Decimal Numbers</b> . . . . .	415
<b>D</b>	<b>Essential Results from Vector and Matrix Algebra</b> . . . . .	417
D.1	Definition of a Vector and a Matrix . . . . .	417
D.2	Properties of Matrices . . . . .	419
D.3	Multiplication, Addition and Subtraction of Matrices . . . . .	420
D.4	The Eigenvalues and Eigenvectors of a Matrix . . . . .	420
D.5	Some Important Matrix, Vector Operations . . . . .	421
D.6	An Orthogonal Matrix – The Concept of Matrix Transpose . . . . .	421
D.7	Diagonalisation of a Matrix . . . . .	422
<b>E</b>	<b>Some Fundamental Material from Probability and Statistics</b> . . . . .	423
E.1	Conditional Probability . . . . .	423
E.2	The Normal Probability Distribution . . . . .	424
E.2.1	The Univariate Case . . . . .	424
E.2.2	The Multivariate Case . . . . .	425

<b>F Penalty Function Derivation of the Maximum Likelihood Decision Rule .....</b>	427
F.1 Loss Functions and Conditional Average Loss .....	427
F.2 A Particular Loss Function .....	428
<b>Subject Index .....</b>	431

# 1

## Sources and Characteristics of Remote Sensing Image Data

### 1.1

#### Introduction to Data Sources

##### 1.1.1

##### Characteristics of Digital Image Data

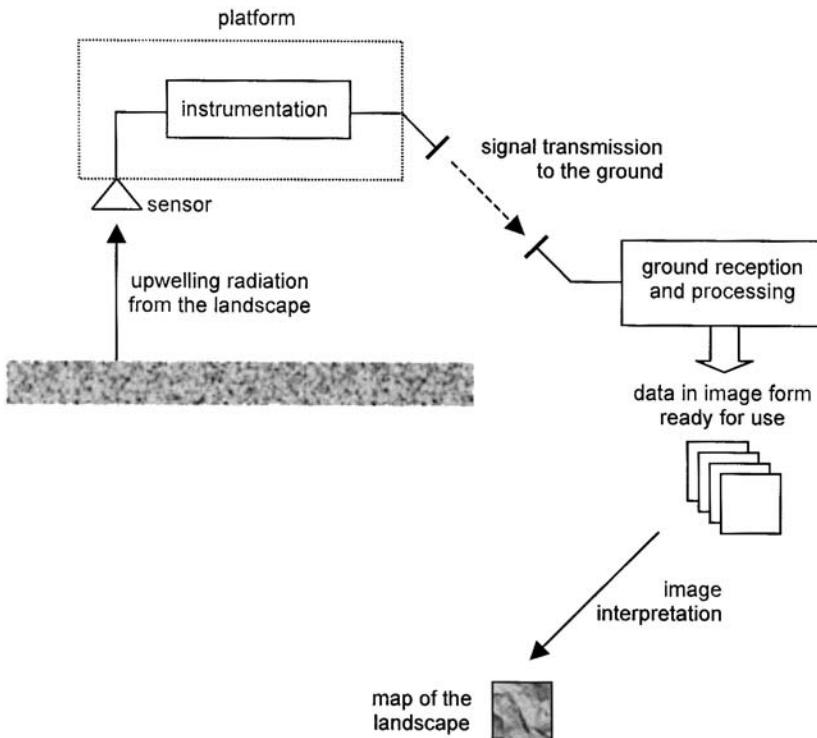
In remote sensing energy emanating from the earth's surface is measured using a sensor mounted on an aircraft or spacecraft platform. That measurement is used to construct an image of the landscape beneath the platform, as depicted in Fig. 1.1.

The energy can be reflected sunlight so that the image recorded is, in many ways, similar to the view we would have of the earth's surface from an aeroplane, although the wavelengths used in remote sensing are often outside the range of human vision. As an alternative, the upwelling energy can be from the earth itself acting as a radiator because of its own temperature. Finally, the energy detected could be scattered from the earth as the result of some illumination by an artificial energy source such as a laser or radar carried on the platform.

Each of these will be outlined in more detail in the following; it is important here to note that the overall system is a complex one involving the scattering or emission of energy from the earth's surface, followed by transmission through the atmosphere to instruments mounted on the remote sensing platform, transmission or carriage of data back to the earth's surface after which it is then processed into image products ready for application by the user. It is really from this point onwards that the material of this book is concerned, viz. we wish to understand how the data, once available in image format, can be used to build maps of features on the landscape.

We generally talk about the imagery recorded as *image data* since it is a primary data source from which we wish to extract usable information. Our ultimate goal is to understand the landscape as imaged and this can be a challenging task involving many of the procedures outlined in this book.

One of the major beneficial characteristics of the image data acquired by sensors on aircraft or spacecraft platforms is that it is readily available in digital format. Spatially the data is composed of discrete picture elements, or *pixels*. Radiometrically

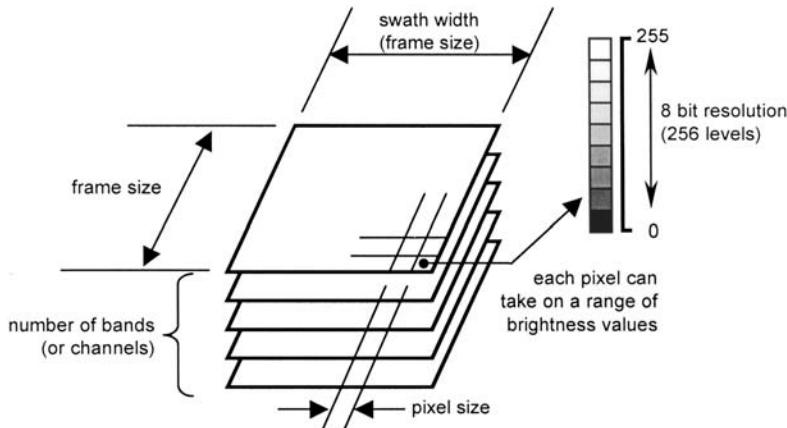


**Fig. 1.1.** Signal and data flow in a remote sensing system

(i.e. in brightness) it is quantised into discrete levels. Even data that is not recorded in digital form initially can be converted into discrete data by the use of digitising equipment. In the early days of remote sensing there was a significant amount of analogue data recorded; now most of the data is available directly in digital form.

The great advantage of having data available digitally is that it can be processed by computer either for machine assisted information extraction or for enhancement of its visual qualities in order to make it more interpretable by a human analyst. Generally, the analyst is referred to as a photointerpreter.

Possibly the most significant characteristic of the image data in a remote sensing system is the wavelength, or range of wavelengths, used in the image acquisition process. If reflected solar radiation is measured images can, in principle, be recorded in the ultraviolet, visible and near-to-middle infrared range of wavelengths. Because of significant atmospheric absorption, ultraviolet measurements are not made. Most common, so-called *optical*, remote sensing systems record data from the visible through to the near and mid infrared range. The energy emitted by the earth itself (dominant in the so-called *thermal* infrared wavelength range) can also be resolved into different wavelengths that help up understand the properties of the earth surface region being imaged.



**Fig. 1.2.** Technical characteristics of digital image data

The visible and infrared range of wavelengths represents only part of the story in remote sensing. We can also image the earth in the microwave range, typical of the wavelengths used in mobile phone, television, FM and radar technologies. While the earth does emit its own level of microwave radiation, it is generally too small to be measured for most remote sensing mapping purposes. Instead, energy is radiated from a platform onto the earth's surface. It is by measuring the energy scattered back to the platform that image data is recorded. Such a system is referred to as *active* since the energy source is provided by the platform. By comparison, remote sensing measurements that depend upon an energy source such as the sun or the earth itself are called *passive*.

From a data handling and analysis point of view the properties of image data of significance are the number and location of the spectral measurements (called spectral bands or channels) provided by a particular sensor, the spatial resolution as described by the pixel size, and the radiometric resolution, as illustrated in Fig. 1.2. The last describes the range and discernible number of discrete brightness values. It is sometimes also referred to as dynamic range and is related to the signal-to-noise ratio of the detectors used. Frequently, the radiometric resolution is expressed in terms of the number of binary digits, or bits, necessary to represent the range of available brightness values. Thus, data with 8 bit radiometric resolution has 256 levels of brightness. Appendix C shows the relationship between radiometric resolution and brightness levels.

Together, the frame size of an image, in equivalent ground kilometres (which is determined by the size of the recorded image swath), the number of spectral bands, the radiometric resolution and the spatial resolution expressed in equivalent ground metres, determine the data volume generated by a particular sensor. That establishes the amount of data to be processed, at least in principle. Consider for example the Landsat Enhanced Thematic Mapper+ (ETM+) instrument. It has seven wavebands with 8 bit radiometric resolution, six of which have 30 m spatial resolution and one of

which has a spatial resolution of 60 m (the thermal band, for which the wavelength is so long that a larger aperture is required to collect sufficient signal energy to maintain the radiometric resolution). An image frame of 185 km × 185 km therefore contains 9.5 million pixels in the thermal band and 38 million pixels in each of the other six bands. At 8 bits per pixel a complete seven band image is composed of  $1.9 \times 10^9$  bits or 1.9 Gbit. Given that one byte is equivalent to 8 bits the data volume would more commonly be expressed as 238 Mbytes.

Appendix A provides an overview of common remote sensing missions and their sensors in terms of the data-related properties of importance to this book. That is useful for indicating orders of magnitude and other properties when determining timing requirements and other figures of merit in assessing image analysis procedures. It also places the analytical material in context with the data gathering phase of remote sensing.

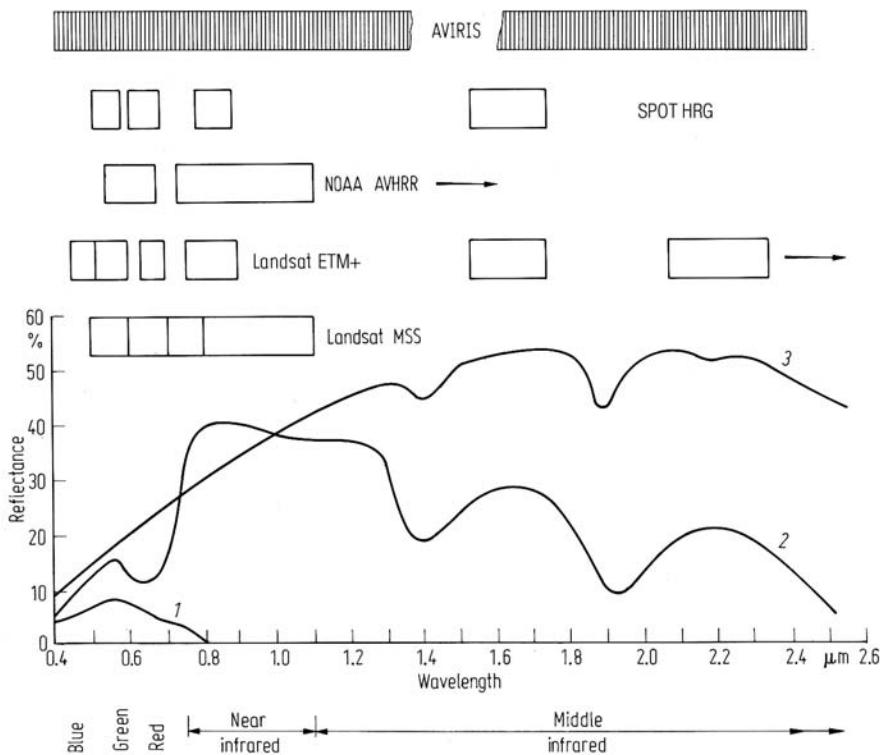
It is of value now to examine the spectral dimension in some detail since the choice of spectral bands for a particular sensor significantly determines the information that can be extracted from the data for a particular application. When more than one spectral measurement is recorded per pixel that data is generally referred to as *multippectral*.

### 1.1.2

#### **Spectral Ranges Commonly Used in Remote Sensing**

In principle, remote sensing systems could measure energy emanating from the earth's surface in any sensible range of wavelengths. However technological considerations, the selective opacity of the earth's atmosphere, scattering from atmospheric particulates and the significance of the data provided exclude certain wavelengths. The major ranges utilized for earth resources sensing are between about 0.4 and 12 µm (the visible/infrared range) and between about 30 to 300 mm (the microwave range). At microwave wavelengths it is often more common to use frequency rather than wavelength to describe ranges of importance. Thus the microwave range of 30 to 300 mm corresponds to frequencies between 1 GHz and 10 GHz. For atmospheric remote sensing, frequencies in the range 20 GHz to 60 GHz are encountered.

The significance of these different ranges lies in the interaction mechanism between the electromagnetic radiation and the materials being examined. In the visible/infrared range the energy measured by a sensor depends upon properties such as the pigmentation, moisture content and cellular structure of vegetation, the mineral and moisture contents of soils and the level of sedimentation of water. At the thermal end of the infrared range it is heat capacity and other thermal properties of the surface and near subsurface that control the strength of radiation detected. In the microwave range, using active imaging systems based upon radar techniques, the roughness of the cover type being detected and its electrical properties, expressed in terms of complex permittivity (which in turn is strongly influenced by moisture content) determine the magnitude of the reflected signal. In the range 20 to 60 GHz, atmospheric oxygen and water vapour have a strong effect on transmission and thus can be inferred by measurements in that range. Thus each range of wavelength has its own strengths



**Fig. 1.3.** Spectral reflectance characteristics of common earth surface materials in the visible and near-to-mid infrared range. 1 Water, 2 vegetation, 3 soil. The positions of spectral bands for common remote sensing instruments are indicated. These are discussed in the following sections

in terms of the information it can contribute to the remote sensing process. Consequently we find systems available that are optimised for and operate in particular spectral ranges, and provide data that complements that from other sensors.

Figure 1.3 depicts how the three dominant earth surface materials of soil, vegetation and water reflect the sun's energy in the visible/reflected infrared range of wavelengths. It is seen that water reflects about 10% or less in the blue-green range, a smaller percentage in the red and certainly no energy in the infrared range. Should the water contain suspended sediments or should a clear water body be shallow enough to allow reflection from the bottom then an increase in apparent water reflection will occur, including a small but significant amount of energy in the near infrared range. This is a result of reflection from the suspension or bottom material.

Soils have a reflectance that increases approximately monotonically with wavelength, however with dips centred at about 1.4 μm, 1.9 μm and 2.7 μm owing to moisture content. These water absorption bands are almost unnoticeable in very dry

soils and sands. In addition, clay soils also have hydroxyl absorption bands at  $1.4\text{ }\mu\text{m}$  and  $2.2\text{ }\mu\text{m}$ .

The vegetation curve is considerably more complex than the other two. In the middle infrared range it is dominated by the water absorption bands at  $1.4\text{ }\mu\text{m}$ ,  $1.9\text{ }\mu\text{m}$  and  $2.7\text{ }\mu\text{m}$ . The plateau between about  $0.7\text{ }\mu\text{m}$  and  $1.3\text{ }\mu\text{m}$  is dominated by plant cell structure while in the visible range of wavelengths it is plant pigmentation that is the major determinant. The curve sketched in Fig. 1.3 is for healthy green vegetation. This has chlorophyll absorption bands in the blue and red regions leaving only green reflection of any significance. This is why we see chlorophyll pigmented plants as green.

An excellent review and discussion of the spectral reflectance characteristics of vegetation, soils, water, snow and clouds can be found in Hoffer (1978) and the Manual of Remote Sensing (1999). This includes a consideration of the physical and biological factors that influence the shapes of the curves, and an indication of the appearances of various cover types in images recorded in different wavelength ranges.

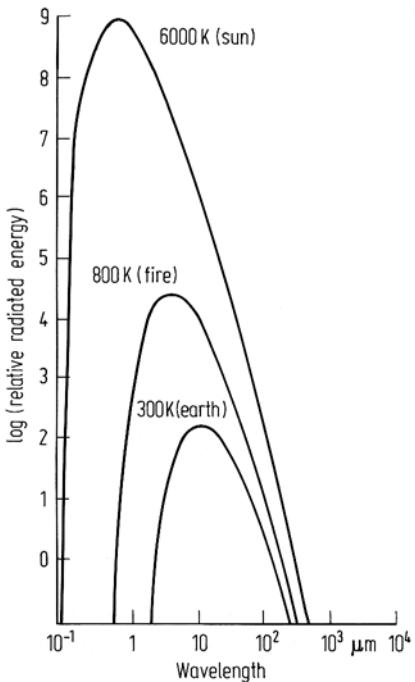
In wavelength ranges between about  $3$  and  $14\text{ }\mu\text{m}$  the level of solar energy actually irradiating the earth's surface is small owing to both the small amount of energy leaving the sun in this range by comparison to the higher levels in the visible and near infrared range (see Fig. 1.4), and the presence of strong atmospheric absorption bands between  $2.6$  and  $3.0\text{ }\mu\text{m}$ ,  $4.2$  and  $4.4\text{ }\mu\text{m}$ , and  $5$  and  $8\text{ }\mu\text{m}$  (Chahine, 1983). Consequently much remote sensing in these bands is of energy being emitted from the earth's surface or objects on the ground rather than of reflected solar radiation.

Figure 1.4 shows the relative amount of energy radiated from perfect black bodies of different temperatures. As seen, the sun at  $6000\text{ K}$  radiates maximally in the visible and near infrared regime but by comparison generates little radiation in the range around  $10\text{ }\mu\text{m}$ . Incidentally, the figure shown does not take any account of how the level of solar radiation is dispersed through the inverse square law process in its travel from the sun to the earth. Consequently if it is desired to compare that curve to others corresponding to black bodies on the earth's surface then it should be appropriately reduced.

The earth, at a temperature of about  $300\text{ K}$  has its maximum emission around  $10$  to  $12\text{ }\mu\text{m}$ . Thus a sensor with sensitivity in this range will measure the amount of heat being radiated from the earth itself. Hot bodies on the earth's surface, such as bushfires, at around  $800\text{ K}$ , have a maximum emission in the range of about  $3$  to  $5\text{ }\mu\text{m}$ . Consequently to map fires, a sensor operating in that range would be used.

Real objects do not behave as perfect black body radiators but rather emit energy at a lower level than that shown in Fig. 1.4. The degree to which an object radiates by comparison to a black body is referred to as its emittance. Thermal remote sensing is sensitive therefore to a combination of an object's temperature and emittance, the last being wavelength dependent.

Microwave remote sensing image data is gathered by measuring the strength of energy scattered back to the satellite or aircraft in response to energy transmitted. The degree of reflection is characterized by the scattering coefficient for the surface



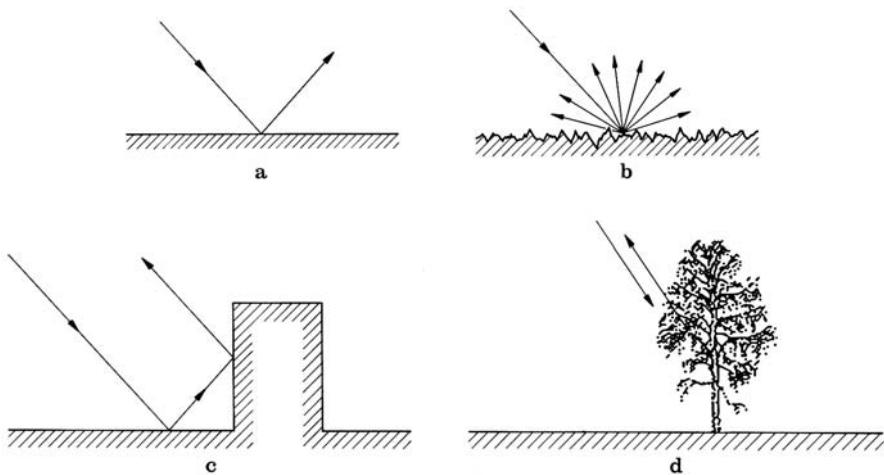
**Fig. 1.4.** Energy from perfect radiators (black bodies) as a function of wavelength

material being imaged. This is a function of the electrical complex permittivity of the material and the roughness of the surface in comparison to a wavelength of the radiation used (Ulaby, Moore & Fung, 1982).

Smooth surfaces act as so-called specular reflectors (i.e. mirror-like) in that the direction of scattering is predominantly away from the incident direction as shown in Fig. 1.5. Consequently they appear dark to black in image data. Rough surfaces act as diffuse reflectors; they scatter the incident energy in all directions as depicted in Fig. 1.5, including back towards the remote sensing platform. As a result they appear light in image data. A third type of surface scattering mechanism is often encountered in microwave image data, particularly associated with manufactured features such as buildings. This is a corner reflector effect, as seen in Fig. 1.5, resulting from the right angle formed between a vertical structure such as a fence, building or ship and a horizontal plane such as the surface of the earth or sea. This gives a very bright response.

Media, such as vegetation canopies and sea ice, exhibit so-called volume scattering behaviour, in that backscattered energy emerges from many, hard to define sites within the volume, as depicted in Fig. 1.5. This leads to a light tonal appearance in radar imagery.

In interpreting image data acquired in the microwave region of the electromagnetic spectrum it is important to recognise that the four reflection mechanisms of Fig. 1.5 are present and modify substantially the tonal differences resulting from sur-



**Fig. 1.5.** **a** Specular, **b** diffuse, **c** corner reflector and **d** volume scattering behaviour, encountered in the formation of microwave image data

face complex permittivity variations. By comparison, imaging in the visible/infrared range in which the sun is the energy source, results almost always from diffuse reflection, allowing the interpreter to concentrate on tonal variations resulting from factors such as those described in association with Fig. 1.3.

A comprehensive treatment of the essential principles of microwave remote sensing will be found in the three volume series by Ulaby, Moore and Fung (1981, 1982, 1985).

### 1.1.3 Concluding Remarks

The purpose of acquiring remote sensing image data is to be able to identify and assess, by some means, surface materials and their spatial properties. Inspection of Fig. 1.3 reveals that cover type identification should be possible if the sensor gathers data at several wavelengths. For example, if for each pixel, measurements of reflection at  $0.65\text{ }\mu\text{m}$  and  $1.0\text{ }\mu\text{m}$  were available (i.e. we had a two band imaging system) then it should be a relatively easy matter to discriminate between the three fundamental cover types based on the relative values in the two bands. For example, vegetation would be bright at  $1.0\text{ }\mu\text{m}$  and very dark at  $0.65\text{ }\mu\text{m}$  whereas soil would be bright in both ranges. Water on the other hand would be black at  $1.0\text{ }\mu\text{m}$  and dull at  $0.65\text{ }\mu\text{m}$ . Clearly if more than two measurement wavelengths were used more precise discrimination should be possible, even with cover types spectrally similar to each other. Consequently remote sensing imaging systems are designed with wavebands that take several samples of the spectral reflectance curves of Fig. 1.3. For each pixel the set of samples can be analysed, either by photointerpretation, or by the automated techniques to be found in Chaps. 8 and 9, to provide a label that associates the pixel with a particular earth surface material.

A similar situation applies when using microwave image data; viz. several different transmission wavelengths can be used to assist in identification of cover types by reason of their different scattering behaviours with wavelength. However a further data dimension is available with microwave imaging owing to the coherent nature of the radiation used. That relates to the *polarizations* of the transmitted and scattered radiation. The polarization of an electromagnetic wave refers to the orientation of the electric field during propagation. For radar systems this can be chosen to be parallel to the earth's surface on transmission (a situation referred to as *horizontal polarization*) or in the plane in which both the incident and scattered rays lie (somewhat inappropriately called *vertical polarisation*). On scattering, some polarization changes can occur and energy can be received as horizontally polarized and/or vertically polarized. The degree of polarization rotation that occurs can be a useful indicator of surface material.

Another consequence of using coherent radiation in radar remote sensing systems, of significance to the interpretation process, is that images exhibit a degree of "speckle". This is a result of constructive and destructive interference of the reflections from surfaces that have random spatial variations of the order of one half a wavelength, or so. Noting that the wavelengths commonly employed in radar remote sensing are between about 30 mm and 300 mm it is usual to find images of most common cover types showing a considerably speckled appearance. Within a homogeneous region for example, such as a crop field, this causes adjacent radar image pixels to have large differences in brightness, a factor which complicates machine-assisted interpretation.

Finally, two radar images recorded over the same region at the same time, or closely spaced in time, can be interfered to allow topographic detail to be revealed. Known as InSAR (for Interferometric Synthetic Aperture Radar) the technique is now widely used for topographic mapping (Zebker and Goldstein, 1986).

## 1.2

### Remote Sensing Platforms

Imaging in remote sensing can be carried out from both satellite and aircraft platforms. In many ways their sensors have similar characteristics although differences in their altitude and stability can lead to very different image properties.

There are essentially two broad classes of satellite program: those satellites that sit at geostationary altitudes above the earth's surface and which are generally associated with weather and climate studies, and those which orbit much closer to the earth's surface and that are generally used for earth surface and oceanographic observations. Usually, the low earth orbiting satellites are in a sun-synchronous orbit, in that their orbital plane precesses around the earth at the same rate that the sun appears to move across the earth's surface. In this manner the satellite acquires data at about the same local time on each orbit.

Low earth orbiting satellites can also be used for meteorological studies. Notwithstanding the differences in altitude, the wavebands used for the geostationary and the

low earth orbiting satellites, and for weather and earth observation satellites, are very comparable. The major distinction in the image data they provide generally lies in the spatial resolutions available. Whereas data acquired for earth resources purposes generally has pixel sizes of less than 100 m, that used for meteorological purposes (both at geostationary and lower altitudes) has a much coarser pixel, often of the order of 1 km.

Appendix A provides detail on the commonly encountered geostationary and low earth orbiting satellite programs over the past four decades or so. Included in that Appendix are also the technical specifications of the data provided by each of their significant imaging instruments.

The imaging technologies utilised in satellite programs have ranged from traditional cameras to mechanical scanners that record images of the earth's surface by moving the instantaneous field of view of the instrument across the earth's surface to record the upwelling energy.

Some weather satellites scan the earth's surface using the spin of the satellite itself while the sensor's pointing direction is varied (at a slower rate) along the axis of the satellite. The image data is then recorded in a raster-scan fashion not unlike that used for the production of television pictures.

A more common image recording mechanism, used in the Landsat program, has been to carry a mechanical scanner that records at right angles to the direction of the satellite motion to produce raster-scans of data. The forward motion of the vehicle then allows an image strip to be built up from the raster-scans. That process is depicted in Fig. 1.6.

More recent technology utilises a "push-broom" mechanism in which a linear imaging array with sufficient detectors is carried on the satellite, normal to the satellite's motion, such that each pixel can be recorded individually. The forward motion of the satellite then allows subsequent pixels to be recorded along the satellite travel direction in the manner shown in Fig. 1.7. As might be expected, the time over which the energy emanating from the earth's surface per pixel is larger with push broom scanning than for the mechanical scanners, generally allowing finer spatial resolutions to be achieved.

Aircraft scanners operate with essentially the same principles as those found on satellites. Both mechanical scanners (often utilising rotating mirrors – see Appendix A) and CCD arrays are commonly employed.

An interesting development in the past decade has been to employ rectangular detector arrays which, in principle, could be used to capture a two dimensional image underneath the satellite. They are normally used, however, to record pixels in the across track direction, as with push broom scanners, with the other dimension employed to record many spectral channels of data simultaneously. This is depicted in Fig. 1.8. Often as many as 200 or so channels are recorded in this manner so that a very good rendition of the spectra depicted in Fig. 1.3 can be obtained. As a result the devices are often referred to as imaging spectrometers and the data described as *hyperspectral*, as against multispectral when of the order of 10 wavebands are recorded. Figure 1.9 shows the quality of the spectral data per pixel possible with an

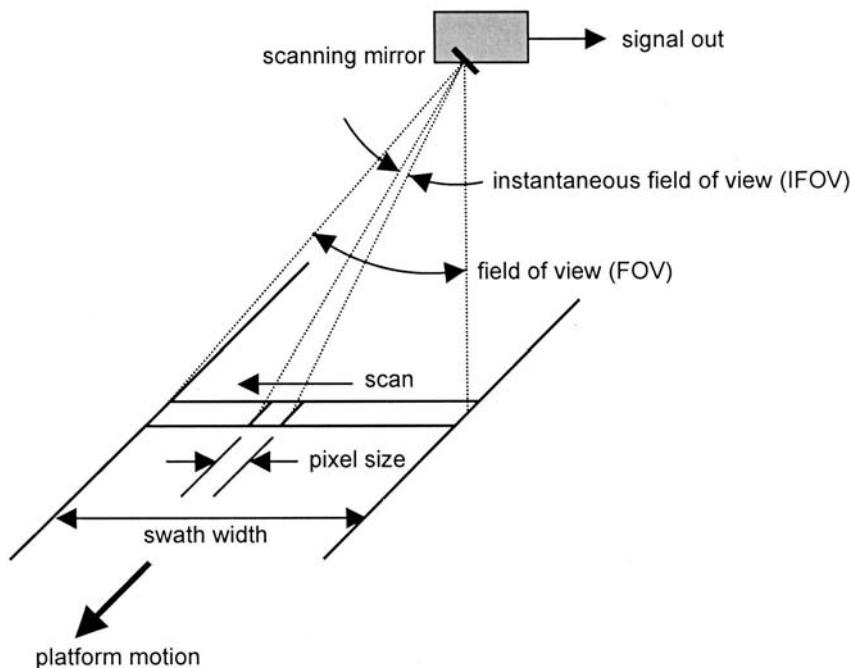


Fig. 1.6. Image formation by mechanical line scanning

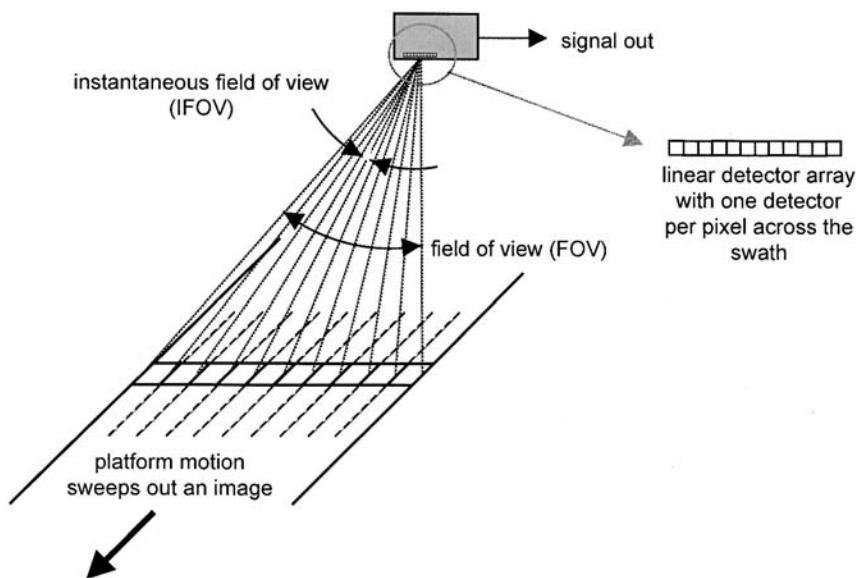
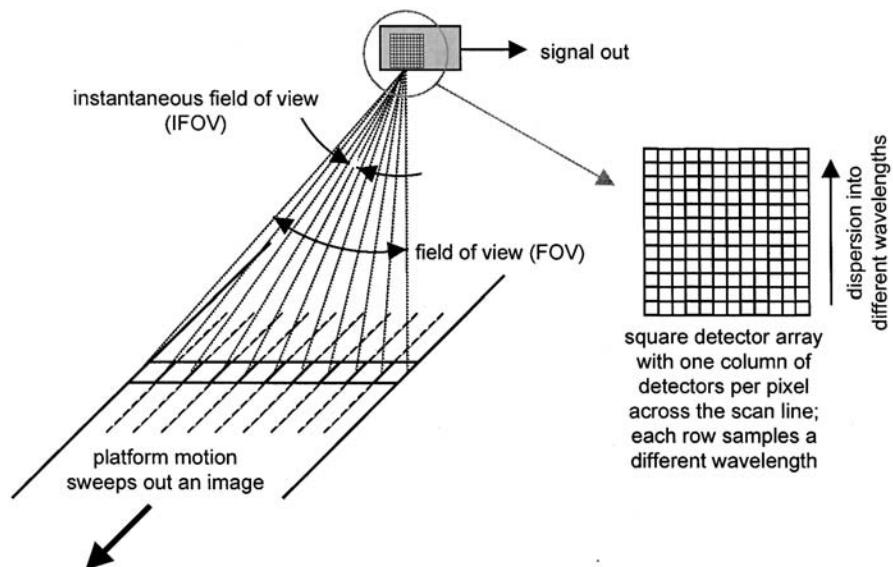


Fig. 1.7. Push broom line scanning in the along-track direction



**Fig. 1.8.** Use of a square detector array to achieve along-track line scanning and the recoding of many spectral measurements simultaneously

imaging spectrometer, compared with the detail obtainable from the Landsat MSS and TM instruments.

## 1.3

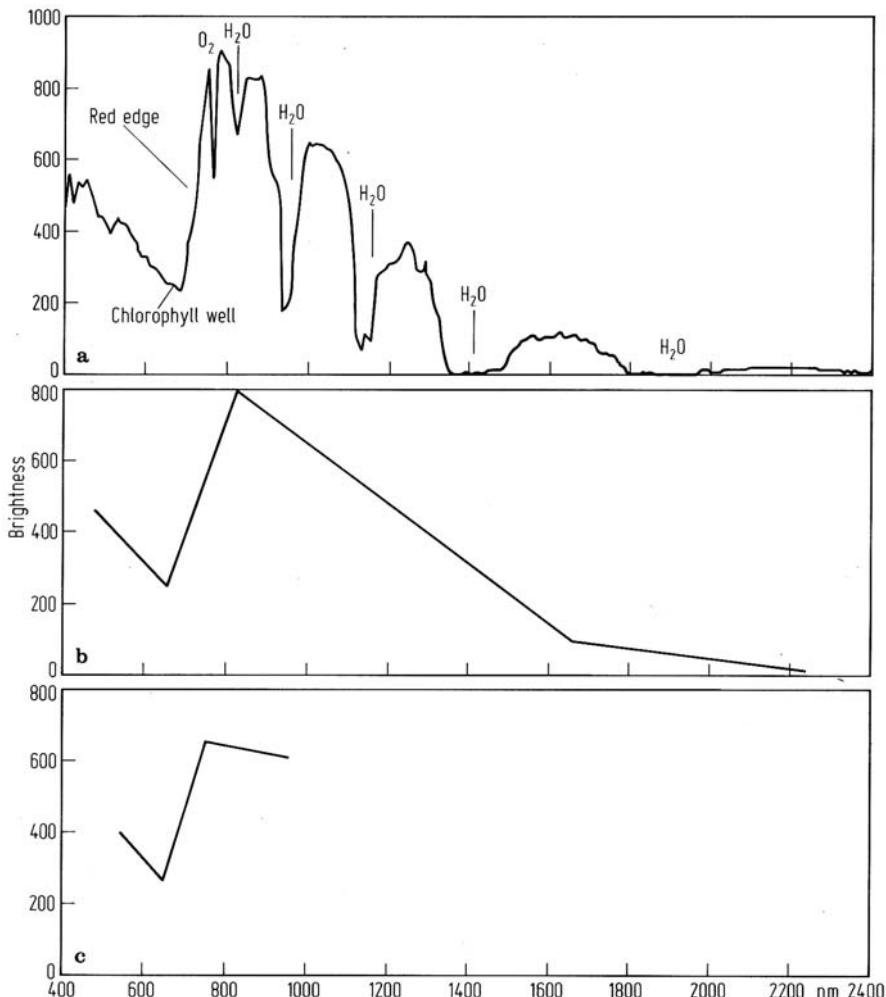
### Image Data Sources in the Microwave Region

#### 1.3.1

##### Side Looking Airborne Radar and Synthetic Aperture Radar

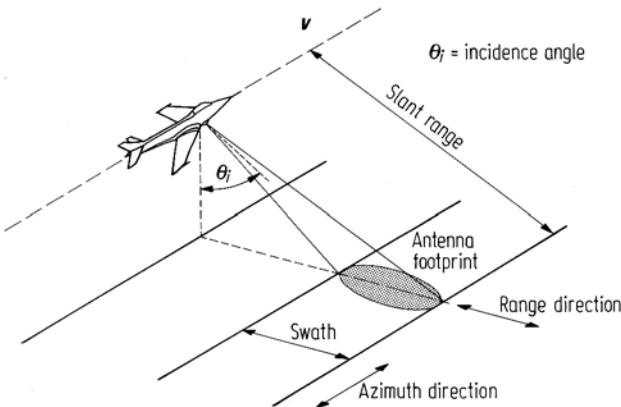
Remote sensing image data in the microwave range of wavelengths is generally gathered using the technique of side-looking radar, as illustrated in Fig. 1.10. When used with aircraft platforms it is more commonly called SLAR (side looking airborne radar), a technique that requires some modification when used from spacecraft altitudes, as discussed in the following.

In SLAR a pulse of electrical energy at the microwave frequency (or wavelength) of interest is radiated to the side of the aircraft at an incidence angle of  $\theta_i$ . By the same principle as radars used for air navigation and shipping, some of this transmitted energy is scattered from the ground and returned to the receiver on the aircraft. The time delay between transmission and reflection identifies the slant distance to the "target" from the aircraft, while the strength of the return contains information on the so-called scattering coefficient of the target region of the earth's surface. The actual received signal from a single transmitted pulse consists of a continuum of



**Fig. 1.9.** Vegetation spectrum recorded by AVIRIS at 10 nm spectral sampling **a**, along with equivalent TM **b** and MSS **c** spectra. In **a** the fine absorption features resulting from atmospheric constituents are shown, along with features normally associated with vegetation spectra.

reflections from the complete region of ground actually illuminated by the radar antenna. In Fig. 1.10 this can be identified as the range beamwidth of the antenna. This is chosen at design to give a relation between swath width and altitude, and tends to be rather broad. By comparison the along-track, or so-called azimuth, beamwidth is chosen as small as possible so that the reflections from a single transmitted pulse can be regarded as having come from a narrow strip of terrain broadside to the aircraft. The forward velocity of the aircraft is then arranged so that the next transmitted pulse illuminates the next strip of terrain along the swath. In this manner the azimuth



**Fig. 1.10.** Principle of side looking radar

beamwidth of the antenna defines the spatial resolution in the azimuth direction whereas the time resolution possible between echos from two adjacent targets in the range direction defines the spatial resolution in the slant direction.

From an image product viewpoint the slant range resolution is not of interest. Rather it is the projection of this onto the horizontal plane as ground range resolution that is of value to the user. A little thought reveals that the ground range resolution is better at larger incidence angles and thus on the far side of the swath; it can be shown that the ground range size of a resolution element (pixel) is given by

$$r_g = c\tau/2 \sin \theta_i$$

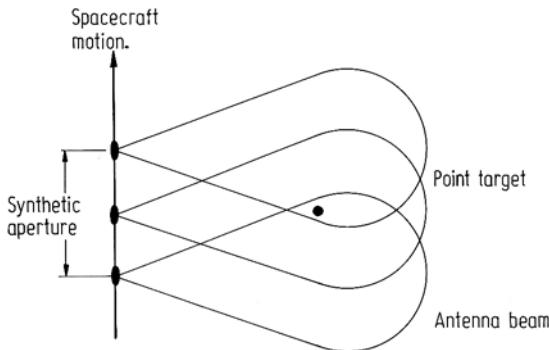
where  $\tau$  is the length of the transmitted pulse and  $c$  is the velocity of light. (Often a simple pulse is not used. Instead a so-called linear chirped waveform is transmitted and signal processing on reception is used to compress this into a narrow pulse. For the present discussion however it is sufficient to consider the transmitted waveform to be a simple pulse or burst of the frequency of interest.)

The azimuth size of a resolution element is related to the length (or aperture) of the transmitting antenna in the azimuth direction,  $l$ , the wavelength  $\lambda$  and the range  $R_0$  between the aircraft and the target, and is given by

$$r_a = R_0\lambda/l$$

This expression shows that a 10 m antenna will yield an azimuth resolution of 20 m at a slant range of 1 km for radiation with a wavelength of 20 cm. However if the slant range is increased to say 100 km – i.e. at low spacecraft altitudes – then a 20 m azimuth resolution would require an antenna of 1 km length, which clearly is impracticable.

Therefore when radar image data is to be acquired from spacecraft, a modification of SLAR referred to as synthetic aperture radar (SAR) is used. Essentially this utilizes the motion of the space vehicle, during transmission of the ranging pulses, to give an effectively long antenna, or a so-called synthetic aperture. This principle is illustrated in Fig. 1.11, wherein it is seen that an intentionally large azimuth beamwidth is



**Fig. 1.11.** The concept of synthesizing a large antenna by utilizing spacecraft motion along its orbital path. Here a view from above is shown, illustrating that a small real antenna is used to ensure a large real beamwidth in azimuth. As a consequence a point on the ground is illuminated by the full synthetic aperture

employed to ensure that a particular spot on the ground is illuminated and thus provides reflections over a length of spacecraft travel equivalent to the synthetic aperture required.

A discussion of the details of the synthetic aperture concept and the signal processing required to produce a high azimuth resolution is beyond the scope of this treatment. The matter is pursued further in Ulaby, Moore and Fung (1982), Elachi et al. (1982), Tomiyasu (1978), and Elachi (1983, 1988).

## 1.4 Spatial Data Sources in General

### 1.4.1 Types of Spatial Data

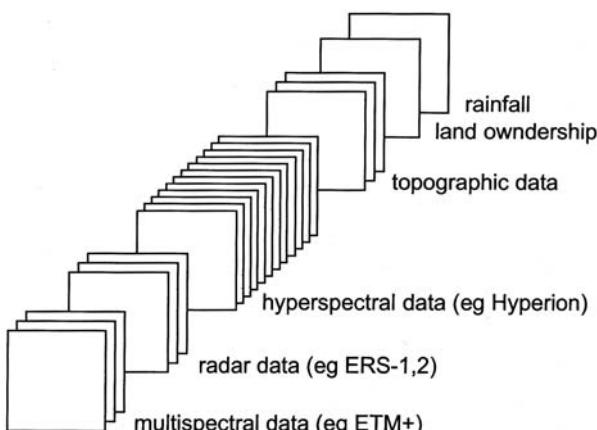
The foregoing sections have addressed sources of multispectral digital image data. Other sources of spatially distributed data are also often available for regions of interest. These include simple maps that show topography, land ownership, roads and the like, through to more specialised sources of spatial data such as maps of geophysical measurements of the area. Frequently these other spatial data sources contain information not available in multispectral imagery and often judicious combinations of multispectral and other map-like data allow inferences to be drawn about regions on the earth's surface not possible when using a single source on its own. Consequently the image analyst ought to be aware of the range of spatial data available for a region and select that subset likely to assist in the information extraction process.

Table 1.1 is an illustration of the range of spatial data one might expect could be available for a given region. This differentiates the data into three types according as to whether it represents point information, line information or area information. Irrespective of type however, for a spatial data set to be manipulated using the techniques

**Table 1.1.** Sources of spatial data

Point	Line	Area
Multispectral data	road maps	land ownership
Topography	powerline grids	town plans
Magnetic measurements	pipeline networks	geological maps
Gravity measurements		land use licenses
Radiometric measurements		land use maps
Rainfall		land cover maps
Geochemistry (in ppm)		soil type maps

of digital image processing it must share two characteristics with multispectral data. First it must be available in discrete form spatially, and in value. In other words it must consist of, or be able to be converted to, pixels with each pixel describing the properties of a given (small) area on the ground: the value ascribed to each pixel must be expressible in digital form. Secondly it must be in correct geographic relation to a multispectral image data set if the two are to be manipulated together. In situations where multispectral data is not used, the pixels in the spatial data source would normally be arranged to be referenced to a map grid. It is usual however, in digital spatial data handling systems, to have all entries in the data set relating to a particular geographical region, mutually registered and referenced to a map base such as the UTM grid system. When available in this manner the data is said to be geocoded. Means by which different data sets can be registered are treated in Sect. 2.5. Such a database is depicted in Fig. 1.12

**Fig. 1.12.** An integrated spatial data source database

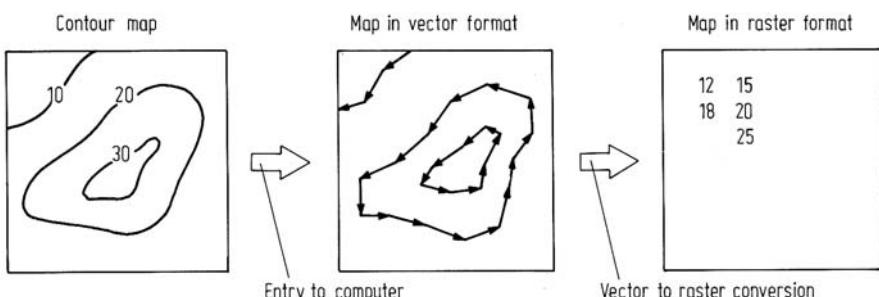
### 1.4.2

#### Data Formats

Not all sources of spatial data are available originally in the pixel oriented digital format depicted in Fig. 1.12. Sometimes the data will be available as analog maps that require digitisation before entry into a digital data base. That is particularly the case with line and area data types, in which case consideration has to be given also to the "value" that will be ascribed to a particular pixel. In line spatial data sources the pixels could be called zero if they were not part of a line and coded to some other number if they formed part of a line of a given type. For a road map, for example, pixels that fall on highways might be given a value of 1 whereas those on secondary roads could be given a value of 2, and so on. On display, the different numbers could be interpreted and output as different colours. In a similar manner numbers can be assigned to different regions when digitizing area spatial data sources.

Conceptually the digitization process may not be straightforward. Consider the case for example, of needing to create a digital topographic map from its analog contour map counterpart. Figure 1.13 illustrates this process. First it is necessary to convert the contours on the paper map to records contained in a computer. This is done by using an input device to mark a series of points on each contour between which the contour is regarded by the computer to be a straight line. Information on a contour at this stage is stored in the computer's memory as a file of points. This is referred to as *vector format* owing to the vectors that can be drawn from point to point (in principle) to reconstruct a contour on a display. Some spatial data handling computer systems operate in vector format entirely. However to be able to exploit the techniques of digital image processing the vector formatted data has to be turned into a set of pixels arranged on rectangular grid centres. This is referred to as *raster format* (or sometimes grid format); the elevation values for each pixel in the raster form are obtained by a process of interpolation over the points recorded on the contours. The operation is referred to as *vector to raster conversion* and is an essential step in entering map data into a digital spatial data base.

Raster format is a natural one for the representation of multispectral image data since data of that type is generated by digitising scanners, is transmitted digitally and



**Fig. 1.13.** Definition of vector and raster format using the illustration of digitising contour data

is recorded digitally. Moreover most image forming devices such as digital cameras operate on a raster display basis, compatible with digital data acquisition and storage. Raster format however is also appealing from a processing point of view since the logical records for the data are the pixel values (irrespective of whether the data is of the point, line or area type) and neighbourhood relationships are easy to establish by means of the pixel addresses. This is important for processing operations that involve near neighbouring groups of pixels. In contrast, vector format does not offer this feature.

### **1.4.3 Geographic Information Systems (GIS)**

The amount of data to be handled in a database that contains spatial sources such as satellite and aircraft imagery along with maps, as listed in Table 1.1, is enormous, particularly if the data covers a large geographical region. Quite clearly therefore thought has to be given to efficient means by which the data types can be stored and retrieved, manipulated, analysed and displayed. This is the role of the geographic information system (GIS). Like its commercial counterpart, the management information system (MIS), the GIS is designed to carry out operations on the data stored in its database, according to a set of user specifications, without the user needing to be knowledgeable about how the data is stored and what data handling and processing procedures are utilized to retrieve and present the information required. Unfortunately because of the nature and volume of data involved in a GIS many of the MIS concepts developed for data base management systems (DBMS) cannot be transferred directly to GIS design although they do provide guidelines. Instead new design concepts have been needed, incorporating the sorts of operation normally carried out with spatial data, and attention has had to be given to efficient coding techniques to facilitate searching through the large numbers of maps and images often involved.

To understand the sorts of spatial data manipulation operations of importance in GIS one must take the view of the resource manager rather than the data analyst. Whereas the latter is concerned with image reconstruction, filtering, transformation and classification, the manager is interested in operations such as those listed in Table 1.2. These provide information from which management strategies and the like can be inferred. Certainly, to be able to implement many, if not most, of these a substantial amount of image processing may be required. However as GIS technology progresses it is expected that the actual image processing being performed would be transparent to the resource manager; the role of the data analyst will then be in part of the GIS design. A good discussion of the essential issues in GIS will be found in Bolstad (2002).

A problem which can arise in image data bases of the type encountered in a GIS is the need to identify one image by reason of its similarity to another. In principle, this could be done by comparing the images pixel-by-pixel; however the computational demand in so doing would be enormous for images of any practical size. Instead effort has been directed to developing codes or signatures for complete images that will allow efficient similarity searching. For example an image histogram could be

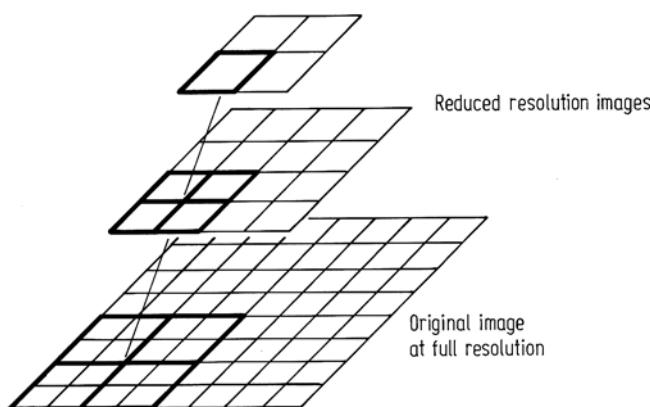
**Table 1.2.** Some GIS data manipulation operations

---

Intersection and overlay of data sets (masking)
Intersection and overlay of polygons (grid cells, etc.) with spatial data
Identification of shapes
Identification of points in polygons
Area determination
Distance determination
Thematic mapping
Proximity calculations (shortest route, etc.)
Search by data
Search by location
Search by user-defined attribute
Similarity searching (e.g. of images)

---

used (see Sect. 4.2); however as geometric detail is not preserved in a histogram this is rarely a suitable code for an image on its own. One effective possibility that has been explored is the use of image pyramids. A pyramid is created by combining groups of pixels in a neighbourhood to produce a new composite pixel of reduced resolution, and thus a low resolution image with fewer pixels. This process is repeated on the processed image to form a new image of lower resolution (and fewer pixels) still. Ultimately the image could be reduced to one single pixel that is a global measure of the image's brightness. Since pixels are combined in neighbourhood groups, spatial detail is propagated up through the pyramid, albeit at decreasing resolution. Figure 1.14 illustrates how an image pyramid is constructed by simple averaging of non-overlapping sets of  $2 \times 2$  pixels. It is a relatively easy matter (see Problem 1.6) to show that the additional memory required to store a complete pyramid, constructed as in the figure, is only 33% more than that required to store just the image itself.

**Fig. 1.14.** Construction of an image pyramid by successively averaging groups of  $2 \times 2$  pixels

Having developed an image pyramid, signatures that can be used to undertake similarity searching include the histograms computed over rows and columns in the uppermost levels of the pyramid (see Problem 1.7). A little thought shows that this allows an enormous number of images to be addressed, particularly if each pixel is represented by an 8 bit brightness value. As a result very fast searching can be carried out on these reduced representations of images.

Image pyramids are discussed by Rosenfeld (1982) and have been considered in the light of image similarity searching by Chien (1980), and data mining by Datcu et al. (2003).

There is sometimes an image processing advantage to be obtained when using a pyramid representation of an image. In edge detection, for example, it is possible to localise edges quickly, without having to search every pixel of an image, by finding apparent edges (regions) in the upper levels of the pyramid. The succeeding lower pixel groupings are then searched to localise the edges better.

Finally the pyramid representation of an image is felt to have some relation to human perception of images. The upper levels contain global features and are therefore not unlike the picture we have when first looking at a scene – generally we take the scene in initially “as a whole” and either miss or ignore detail. Then we focus on regions of interest for which we pay attention to detail because of the information it provides us with.

#### 1.4.4

#### The Challenge to Image Processing and Analysis

Much of the experience gained with digital image processing and analysis in remote sensing has been with multispectral image data. In principle however any spatial data type in digital format can be processed using the techniques and procedures presented in this book. Information extraction from geophysical data could be facilitated, for example, if a degree of sharpening is applied prior to photointerpretation, while colour density slicing could assist the interpretation of topography. However the real challenge to the image analyst arises when data of mixed types are to be processed together. Several issues warrant comment.

The first relates to differences in resolution, an issue that arises also when treating multi-source satellite data such as Landsat ETM+ and Aqua MODIS. The analyst must decide, for example what common pixel size will be used when co-registering the data, since either resolution or coverage will normally be sacrificed. Clearly this decision will be based on the needs of a particular application and is a challenge more to the analyst than the algorithms.

The more important consideration however is in relation to techniques for machine assisted interpretation. There is little doubt that combined multispectral and, say, topographic or land ownership maps can yield more precise thematic (i.e. category of land cover, etc.) information for a particular region than the multispectral data on its own. Indeed the combination of these sources is often employed in photointerpretive studies.

The issue is complicated further when it is recalled that much of the non-spectral, spatial data available is not in numerical point form but rather is in nominal area or line format. With these, image analysis algorithms developed algebraically will not be suitable. Rather same degree of logical processing of labels combined with algebraic processing of arithmetic values (such as pixel brightnesses) is necessary.

Chapter 12 addresses this issue by considering several numerical and knowledge-based image analysis methods, which lend themselves to handling both numerical and non-numerical data sources.

## 1.5

### A Comparison of Scales in Digital Image Data

Because of IFOV differences the digital images provided by various remote sensing sensors will find application at different scales. As a guide Table 1.3 relates scale to spatial resolution; this has been derived somewhat simplistically by considering an image pixel to be too coarse if it approaches 0.1 mm in size on a photographic product at a given scale. Thus Landsat MSS data is suggested as being suitable for scales smaller than about 1 : 500,000 whereas NOAA AVHRR data is suitable for scales below 1 : 10,000,000. Detailed discussions of image quality in relation to scale will be found in Welch (1982), Forster (1985), Woodcock and Strahler (1987) and Light (1990).

**Table 1.3.** Suggested maximum scales of photographic products as a function of effective ground pixel size (based on 0.1 mm printed pixel)

Scale	Approx. Pixel Size (m)	Sensor (nominal)
1 : 10,000	1	Ikonos panchromatic
1 : 50,000	5	aircraft MSS, Ikonos XS
1 : 100,000	10	Spot HRG
1 : 250,000	25	Spot HRVIR, Landsat TM
1 : 500,000	50	Landsat TM, LISS
1 : 5,000,000	500	OCTS, OCM
1 : 10,000,000	1000	NOAA AVHRR, MODIS
1 : 50,000,000	5000	GMS thermal IR band

## References for Chapter 1

More details on satellite programs, along with information on sensors and data characteristics can be found in the web sites of the responsible agencies. Some of particular use are:

*For weather satellites*

<http://www.wmo.ch>  
<http://www.ncdc.noaa.gov>  
<http://www.eumetsat.de>

*For earth observation satellites*

<http://www.nasa.gov>  
<http://www.orbimage.com>  
<http://hdsn.eoc.nasda.go.jp>  
<http://www.spotimage.fr>  
<http://www.spaceimaging.com>  
<http://earth.esa.int>  
<http://www.isro.org>

*For radar missions*

<http://www.rsi.ca>  
<http://southport.jpl.nasa.gov>

*For imaging spectrometers*

<http://www.techexpo.com/WWW/pto-knowledge/>

The Manual of Remote Sensing (1999) provides an excellent and comprehensive coverage of the field of remote sensing, and spectral reflectance characteristics in particular.

Cloude et al. (1998) gives a contemporary account of interferometric synthetic aperture radar and its application to topographic mapping.

- R. Bolstad, 2002: GIS Fundamentals: A First Text on Geographical Information Systems, Eider.
- M.T. Chahine, 1983: Interaction Mechanisms within the Atmosphere. In Manual of Remote Sensing, R.N. Colwell (Ed). 2e. American Society of Photogrammetry, Falls Church, Va.
- Y.T. Chien, 1980: Hierarchical Data Structures for Picture Storage, Retrieval and Classification. In Pictorial Informations Systems, S.K. Chang and K.S. Fu (Eds.), Springer-Verlag, Berlin.
- S.R. Cloude and K.P. Papathanassiou, 1998: Polarimetric SAR Interferometry. IEEE Trans. Geoscience and Remote Sensing, 36, 1551–1565.
- M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A. Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P. Marchetti and S. D'Elia, 2003: Information Mining in Remote Sensing Image Archives. IEEE Trans. Geoscience and Remote Sensing, 41, 2923–2936.
- C. Elachi (Chairman), 1983: Spaceborne Imaging Radar Symposium. Jet Propulsion Laboratory, January 17–20. JPL Publication 83–11.
- C. Elachi, T. Bicknell, R.L. Jordan and C. Wu, 1982: Spaceborne Synthetic Aperture Imaging Radars. Applications, Techniques and Technology. Proc. IEEE, 70, 1174–1209.

- C. Elachi, 1988: Spaceborne Radar Remote Sensing: Applications and Techniques. N.Y., IEEE.
- B.C. Forster, 1985: Mapping Potential of Future Spaceborne Remote Sensing Systems. Institution of Surveyors (Australia) Annual Congress, Alice Springs.
- R.M. Hoffer, 1978: Biological and Physical Considerations in Applying Computer-Aided Analysis Techniques to Remote Sensor Data. In P.H. Swain and S.M. Davis, Eds., Remote Sensing: The Quantitative Approach, N.Y., McGraw-Hill.
- D.L. Light, 1990: Characteristics of Remote Sensors for Mapping and Earth Science Applications. Photogrammetric Engineering and Remote Sensing, 56, 1613–1623.
- Manual of Remote Sensing, Remote Sensing for the Earth Sciences, 1999. A.N. Renez and R.A Ryerson (Eds.), 3rd ed., NY, Wiley.
- A. Rosenfeld, 1982: Quadtrees and Pyramids: Hierarchical Representation of Images, Report TR-1171, Computer Vision Laboratory, University of Maryland.
- K. Tomiyasu, 1978: Tutorial Review of Synthetic-Aperture Radar (SAR) with Applications to Imaging of the Ocean Surface. Proc. IEEE, 66, 563–583.
- R. Welch, 1982: Image Quality Requirements for Mapping from Satellite Data. Proc. Int. Soc. Photogrammetry and Remote Sensing, Commission 1. Primary Data Acquisition, Canberra.
- F.T. Ulaby, R.K. Moore and A.K. Fung, 1981, 1982, 1985: Microwave Remote Sensing, Active and Passive. Vols. 1,2,3 Reading Mass. Addison-Wesley.
- C.E. Woodcock and A.H. Strahler, 1987: The Factor of Scale in Remote Sensing. Remote Sensing of Environment, 21, 311–332.
- H.A. Zebker and R.M. Goldstein, 1986: Topographic Mapping from Interferometric Synthetic Aperture Radar Observations. J. Geophysical Research, 91, 4993–4999.

## Problems

**1.1** Plot graphs of pixel size in equivalent ground metres as a function of angle from nadir across a swath for

- Landsat MSS with IFOV of 0.086 mrad, FOV =  $11.56^\circ$ ,
- NOAA AVHRR with IFOV = 1.3 mrad, FOV = 2700 km, altitude = 833 km,
- an aircraft scanner with IFOV = 2.5 mrad, FOV =  $80^\circ$  flying at 1000 m AGL (above ground level),

producing separate graphs for the along track and across track dimensions of the pixel. Replot the graphs to indicate pixel size relative to that at nadir.

**1.2** Imagine you have available image data from a multispectral scanner that has two narrow spectral bands. One is centred on  $0.65\text{ }\mu\text{m}$  and the other on  $1.0\text{ }\mu\text{m}$  wavelength. Suppose the corresponding region on the earth's surface consists of water, vegetation and soil.

Construct a graph with two axes, one representing the brightness of a pixel in the  $0.65\text{ }\mu\text{m}$  band and the other representing the brightness of the pixel in the  $1.0\text{ }\mu\text{m}$  band. In this show where you would expect to find vegetation pixels, soil pixels and water pixels. Note how straight lines could, in principle, be drawn between the three groups of pixels so that if a computer had the equations of these lines stored in its memory it could use them to identify every pixel in the image.

Repeat the exercise for a scanner with bands centred on  $0.95\text{ }\mu\text{m}$  and  $1.05\text{ }\mu\text{m}$ .

**1.3** There are 460  $185\text{ km} \times 185\text{ km}$  frames of Landsat data that cover Australia. Compute the daily data rate (in Gbit/day) for Australia provided by the ETM+ sensor on Landsat 7, assuming all possible scenes are recorded.

**1.4** Assume a “frame” of image data consists of a segment along the track of the satellite, as long as the swath is wide. Compute the data volume of a single frame from each of the following sensors and produce a graph of average data volume per wavelength band versus pixel size.

NOAA AVHRR

Aqua MODIS

ADEOS AVNIR (multispectral)

Landsat ETM+

Spot HRG (multispectral)

**1.5** Determine a relationship between swath width and orbital repeat cycle for a polar orbiting satellite at an altitude of 800 km, assuming that adjacent swaths overlap by 10% at the equator.

**1.6** An image pyramid is to be constructed in the following manner: Groups of  $2 \times 2$  pixels are averaged to form single pixels and thereby reduce the number of pixels in the image by a factor of 4, while reducing its resolution as well. Groups of  $2 \times 2$  pixels in the reduced resolution image are then averaged to form a third version of lower resolution still. This process can be continued until the original image is represented by a pyramid of progressively lower resolution images with a single pixel at the top.

Determine the additional memory required to store the complete pyramid by comparison to storing just the image itself. (Hint: Use the properties of a geometric progression.)

Repeat the exercise for the case of a pyramid built by averaging  $3 \times 3$  groups of pixels.

**1.7** A particular image data base is to be constructed to allow similarity searching to be performed on sets of binary images i.e. on images in which pixels take on brightness values of 0 or 1 only. Image pyramids are to be stored in the data base where each succeeding higher level in a pyramid has pixels derived from  $3 \times 3$  groups in the immediately lower level. The value of the pixel in the higher level is to be that of the majority of pixels in the corresponding lower group. The uppermost level in the pyramid is a  $3 \times 3$  image.

- (i) How much additional storage is required to store the pyramids rather than just the original images?
- (ii) The search algorithm to be implemented on the top level of the pyramid is to consist of histogram comparison. In this histograms are taken of the pixels along each row and down each column and a pair of images are ‘matched’ when all of these histograms are the same for both images. In principle, how many distinct images can be addressed using the top level only?
- (iii) An alternative search algorithm to that mentioned in (ii) is to compute just the simple histogram of all the pixels in the top level of the pyramid. How many distinct images could be addressed in this case using the top level only?
- (iv) Would you recommend storing the complete pyramid for each image or just the original image plus histogram information for the upper levels of a pyramid?
- (v) An alternative means by which the upper levels of the pyramid could be coded is simply by counting and storing the fraction of 1’s which occurs in each of the first few uppermost levels. Suppose this is done for the top three levels. Show how a feature or pattern space could be constructed for the complete image data base, using the 1’s fractions for the upper levels in each image, which can then be analysed and searched using pattern classification procedures.

**1.8** A particular satellite carries a high resolution optical sensor with 1 m spatial resolution and is at 800 km altitude in a near polar orbit. Orbital period is related to orbital radius by:

$$T = 2\pi \sqrt{\frac{r^3}{\mu}}$$

where  $\mu = 3.986 \times 10^{14} m^3 s^{-2}$ , and orbital radius is given by

$$r = a + h$$

in which  $a = 6.378$  Mm and  $h$  is altitude.

If the orbit is arranged such that complete earth coverage is possible, how long will that take if there are 2048 pixels per swath? Consequently, what sorts of applications would such a satellite be used for?

**1.9** Suppose a particular sensor recorded reflectance data in just two wavebands. Further, suppose its radiometric resolution were only 2 bits – i.e. are just four levels of grey available in each of the two bands. What is the theoretical maximum number of different cover types that could be discriminated with the sensor – i.e. how many different unique brightness value-waveband pairs are available? Those pairs are in fact the individually resolvable sites in the coordinate space discussed in problem 1.2.

Show that if a sensor has  $c$  channels and a radiometric resolution of  $b$  bits that the total number of sites in the space is  $2^{bc}$ . How many different sites are there for the following sensors?

Spot HRV

Landsat Thematic Mapper

OrbView2 SeaWiFS

Aqua MODIS

EO-1 Hyperion.

For an image of  $512 \times 512$  pixels how many sites, on the average, will be occupied for each of the sensors above?

## **2**

# **Error Correction and Registration of Image Data**

When image data is recorded by sensors on satellites and aircraft it can contain errors in geometry and in the measured brightness values of the pixels. The latter are referred to as radiometric errors and can result from the instrumentation used to record the data, from the wavelength dependence of solar radiation and from the effect of the atmosphere. Image geometry errors can arise in many ways. The relative motions of the platform, its scanners and the earth, for example, can lead to errors of a skewing nature in an image product. Non-idealities in the sensors themselves, the curvature of the earth and uncontrolled variations in the position and attitude of the remote sensing platform can all lead to geometric errors of varying degrees of severity.

When an image is to be utilized it is frequently necessary to make corrections in brightness and geometry if the accuracy of interpretation, either manually or by machine, is not to be prejudiced. For many applications only the major sources of error will require compensation whereas in others more precise correction will be necessary.

It is the purpose of this chapter to discuss the nature of the radiometric and geometric errors commonly encountered in remote sensing images and to develop computational procedures that are used for their compensation. While this is the principal intention, the procedures to be presented find more general application as well, such as in registering together sets of images of the same region but at different times, and in performing operations such as scale changing and zooming (magnification).

Radiometric correction procedures for hyperspectral imagery are treated separately in Chap. 13.

### **2.1**

#### **Sources of Radiometric Distortion**

Mechanisms that affect the measured brightness values of the pixels in an image can lead to two broad types of radiometric distortion. First, the relative distribution of

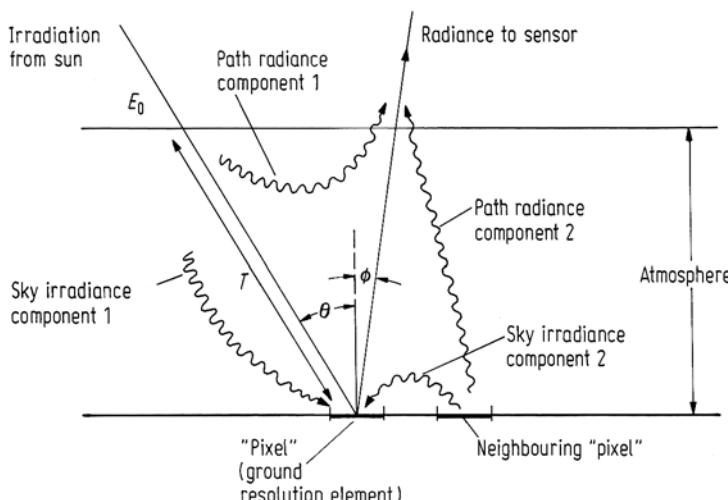
brightness over an image in a given band can be different to that in the ground scene. Secondly, the relative brightness of a single pixel from band to band can be distorted compared with the spectral reflectance character of the corresponding region on the ground. Both types can result from the presence of the atmosphere as a transmission medium through which radiation must travel from its source to the sensors, and can be a result also of instrumentation effects.

### 2.1.1

#### The Effect of the Atmosphere on Radiation

Figure 2.1 depicts the effect the atmosphere has on the measured brightness value of a single pixel for a passive remote sensing system in which the sun is the source of energy, as in the visible and reflective infrared regions. In the absence of an atmosphere the signal measured by the sensor will be a function simply of the level of energy from the sun, actually incident on the pixel, and the reflectance properties of the pixel itself. However the presence of the atmosphere can modify the situation significantly as depicted in the diagram. Before discussing this in detail it is of value to introduce some definitions of radiometric quantities as these will serve to simplify explanations and will allow correction equations to be properly formulated.

Imagine the sun as a source of energy emitting at a given rate of Joules per second, or Watts. This energy radiates through space isotropically in an inverse square law fashion so that at a given distance the sun's emission can be measured as Watts per square metre (given as the power emitted divided by the surface area of a sphere at that distance). This power density is called *irradiance*, a property that can be used to describe the strength of any emitter of electromagnetic energy.



**Fig. 2.1.** The effect of the atmosphere in determining various paths for energy to illuminate a (equivalent ground) pixel and to reach the sensor

We can measure a level of solar irradiance at the earth's surface. If the surface is perfectly diffuse then this amount is scattered uniformly into the upper hemisphere. The amount of power density scattered in a particular direction is defined by its density per solid angle, since equal amounts are scattered into equal cones of solid angle. This quantity is called *radiance* and has units of Watts per square metre per steradian ( $\text{Wm}^{-2}\text{sr}^{-1}$ ).

The emission of energy by bodies such as the sun is wavelength dependent, as seen in Fig. 1.4, so that often the term *spectral irradiance* is used to describe how much power density is available incrementally across the wavelength range. Spectral irradiance is typically measured in  $\text{Wm}^{-2}\mu\text{m}^{-1}$ .

As an illustration of how these quantities might be used suppose, in the absence of atmosphere, the solar spectral irradiance at the earth is  $E_\lambda$ . If the solar zenith angle (measured from the normal to the surface) is  $\theta$  as shown in Fig. 2.1 then the spectral irradiance (spectral power density) on the earth's surface is  $E_\lambda \cos \theta$ . This gives an available irradiance between wavelengths  $\lambda_1$  and  $\lambda_2$  of

$$E_{os} = \int_{\lambda_1}^{\lambda_2} E_\lambda \cos \theta d\lambda. \quad \text{Wm}^{-2}$$

In remote sensing the wavebands used ( $\Delta\lambda = \lambda_2 - \lambda_1$ ) are frequently narrow enough to assume

$$E_{os} = E_{\Delta\lambda} \cos \theta \Delta\lambda \quad \text{Wm}^{-2} \quad (2.1)$$

where  $E_{\Delta\lambda}$  is the average spectral irradiance in the band  $\Delta\lambda$ .

Suppose the surface has a reflectance  $R$ . This describes what proportion of the incident energy is reflected. If the surface is diffuse then the radiance scattered into the upper hemisphere and available for measurement is

$$L = E_{\Delta\lambda} \cos \theta \Delta\lambda R / \pi \quad \text{Wm}^{-2}\text{sr}^{-1} \quad (2.2)$$

where the divisor  $\pi$  accounts for the upper hemisphere of solid angle. Knowing  $L$  it is possible to determine the power detected by a sensor, and the digital count value (or grey level) given in the digital data product from a particular sensor which is directly related to the radiance of the scene. If we call the digital value (between 0 and 255 for example)  $C$ , then the measured radiance of a particular pixel is

$$L = Ck + L_{min} \quad \text{Wm}^{-2}\text{sr}^{-1} \quad (2.3)$$

where  $k = (L_{max} - L_{min})/C_{max}$  in which  $L_{max}$  and  $L_{min}$  are the maximum and minimum measurable radiances of the sensor. These are usually available from the sensor manufacturer or operator.

Equation (2.2) relates to the ideal case of no atmosphere. When an atmosphere is present there are several mechanism that must be taken into account that modify (2.2). These are a result of scattering and absorption by the particles in the atmosphere.

Absorption by atmospheric molecules is a selective process that converts incoming energy into heat. In particular, molecules of oxygen, carbon dioxide, ozone and water attenuate the radiation very strongly in certain wavebands. Sensors commonly used in solid earth and ocean remote sensing are usually designed to operate away

from these regions so that the effects are small. Scattering by atmospheric particles is then the dominant mechanism that leads to radiometric distortion in image data (apart from sensor effects).

There are two broadly identified scattering mechanisms. The first is scattering by the air molecules themselves. This is called Rayleigh scattering and is an inverse fourth power function of the wavelength used. The other is called aerosol or Mie scattering and is a result of scattering of the radiation from larger particles such as those associated with smoke, haze and fumes. These particulates are of the order of one tenth to ten wavelengths. Mie scattering is also wavelength dependent, although not as strongly as Rayleigh scattering. When the atmospheric particulates become much larger than a wavelength, such as those common in fogs, clouds and dust, the wavelength dependence disappears.

In a clear ideal atmosphere Rayleigh scattering is the only mechanism present. It accounts, for example, for the blueness of the sky. Because the shorter (blue) wavelengths are scattered more than the longer (red) wavelengths we are more likely to see blue when looking in any direction in the sky. Likewise the reddish appearance of sunset is also caused by Rayleigh scattering. This is a result of the long atmospheric path the radiation has to follow at sunset during which most short wavelength radiation is scattered away from direct line of sight by comparison to the longer wavelengths.

In contrast to Rayleigh scattering, fogs and clouds appear white or bluish-white owing to the (near) non-selective scattering caused by the larger particles.

We are now in the position to appreciate the effect of the atmosphere on the radiation that ultimately reaches a sensor. We will do this by reference to Fig. 2.1, commencing with the incoming solar radiation. The effects are identified by name:

*Transmittance.* In the absence of atmosphere transmittance is 100%. However because of scattering and absorption not all of the available solar irradiance reaches the ground. The amount that does, relative to that for no atmosphere, is called the transmittance. Let this be called  $T_\theta$  the subscript indicating its dependence on the zenith angle of the source because of the longer path length through the atmosphere. In a similar way there is an atmospheric transmittance  $T_\theta$  to be taken into account between the point of reflection and the sensor.

*Sky irradiance.* Because the radiation is scattered on its travel down through the atmosphere a particular pixel will be irradiated both by energy on the direct path in Fig. 2.1 and also by energy scattered from atmospheric constituents. The path for the latter is undefined and in fact diffuse. A pixel can also receive some energy that has been reflected from surrounding pixels and then, by atmospheric scattering, is again directed downwards. This is the sky irradiance component 2 identified in Fig. 2.1. We will call the sky irradiance at the pixel  $E_D$ .

*Path radiance.* Again because of scattering alone, radiation can reach the sensor from adjacent pixels and also via diffuse scattering of the incoming

radiation that is actually scattered towards the sensor by the atmospheric constituents before it reaches the ground. These two components are referred to as path radiance and denoted  $L_p$ .

Having defined these effects we are now in the position to determine how the radiance measured by the sensor is affected by the presence of the atmosphere. First the total irradiance at the earth's surface now becomes, instead of (2.1)

$$E_G = E_{\Delta\lambda} T_\theta \cos \theta \Delta\lambda + E_D \quad \text{Wm}^{-2}$$

where, for simplicity, it has been assumed that the diffuse sky irradiance is not a function of wavelength (in the waveband of interest). The radiance therefore due to this global irradiance of the pixel becomes

$$L_T = \frac{R}{\pi} \{E_{\Delta\lambda} T_\theta \cos \theta \Delta\lambda + E_D\} \quad \text{Wm}^{-2} \text{sr}^{-1}$$

Above the atmosphere the total radiance available to the sensor then becomes

$$L_s = \frac{RT_\phi}{\pi} \{E_{\Delta\lambda} T_\theta \cos \theta \Delta\lambda + E_D\} + L_p \quad \text{Wm}^{-2} \text{sr}^{-1} \quad (2.4)$$

It is this quantity therefore that should be used in (2.3) to relate the digital count value to measured radiance.

### 2.1.2

#### Atmospheric Effects on Remote Sensing Imagery

A result of the scattering caused by the atmosphere is that fine detail in image data will be obscured. Consequently it is important in applications where one is dependent upon the limit of sensor resolution available, such as in urban studies, to take steps to correct for atmospheric effects.

It is important also to consider carefully the effects of the atmosphere on remote sensing systems with wide fields of view in which there will be an appreciable difference in atmospheric path length between nadir and the extremities of the swath. This will be of significance for example with aircraft scanners and satellite missions such as NOAA.

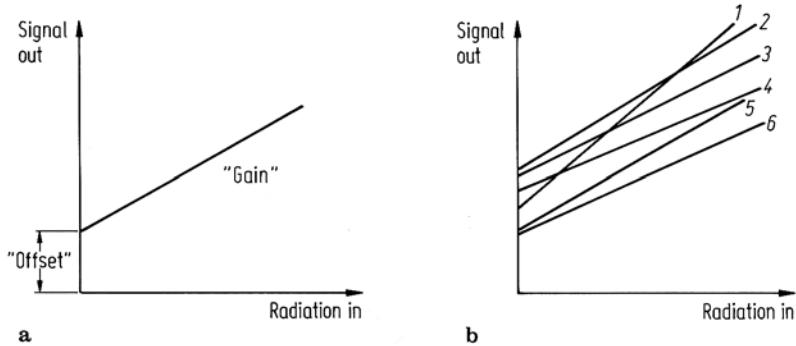
Finally, and perhaps most importantly, because both Rayleigh and Mie scattering are wavelength dependent the effects of the atmosphere will be different in the different wavebands of a given sensor system. In the case of the Landsat Thematic Mapper the visible blue band (0.45 to 0.52  $\mu\text{m}$ ) can be affected appreciably by comparison to the middle infrared band (1.55 to 1.75  $\mu\text{m}$ ). This leads to a loss in calibration of the set of brightnesses associated with a particular pixel.

Methods for correcting for the radiometric distortion caused by the atmosphere are discussed in Sect. 2.2.

### 2.1.3

#### Instrumentation Errors

Radiometric errors within a band and between bands can also be caused by the design and operation of the sensor system. Band to band errors from this source are normally



**Fig. 2.2. a** Transfer characteristic of a radiation detector; **b** Hypothetical mismatches in detector characteristics in the same band

ignored by comparison to band to band errors from atmospheric effects. However errors within a band can be quite severe and often require correction to render an image product useful.

The most significant of these errors is related to the detector system. An ideal radiation detector should have a transfer characteristic (radiation in, signal out) as shown in Fig. 2.2a. This should be linear so that there is a proportional increase and decrease of signal with detected radiation level. Real detectors will have some degree of nonlinearity (which is ignored here) and will also give a small signal out even when no radiation is being detected. Historically this is known as dark current and is related to the residual electronic noise in the system at any temperature above absolute zero; we will call it an "offset". The slope of the characteristic is frequently called its transfer gain or just simply "gain".

Most remote sensors involve a multitude of detectors. In the case of the Landsat MSS there were 6 per band, for the Landsat TM there are 16 per band and for the SPOT HRV there are 6000 in the panchromatic mode of operation. Each of these detectors will have slightly different transfer characteristics as described by their gains and offsets, as shown in Fig. 2.2b.

In the case of scanners such as the TM and MSS these imbalances will lead to striping in the across swath direction as shown in Fig. 2.4a. For the HRV longitudinal striping may occur.

## 2.2

### Correction of Radiometric Distortion

In contrast to geometric correction, in which all sources of error are often rectified together, radiometric correction procedures must be specific to the nature of the distortion.

### 2.2.1

#### Detailed Correction of Atmospheric Effects

Rectifying image data to remove as much as possible the degrading effects of the atmosphere entails modelling the scattering and absorption processes that take place and establishing how these determine both the transmittances of the various paths and the different components of sky irradiance and path radiance. When available these can be used in (2.3) and (2.4) to relate the digital count values given for the pixels in each band of data  $C$ , to the true reflectance  $R$  of the surface being imaged. An example of how this can be done is given by Forster (1984) for the case of Landsat MSS data; Forster also gives source material and tables to assist in the computations. Some aspects of this example are given here to establish relative quantities.

Forster considers the case of a Landsat 2 MSS image in the wavelength range 0.8 to 1.1  $\mu\text{m}$  acquired at Sydney, Australia on 14 December 1980 at 9:05 a.m. local time. At the time of overpass the atmospheric conditions were

temperature	29°C	
relative humidity	24%	
atmospheric pressure	1004 mbar	measured at 30 m above sea level.
visibility	65 km	

Based upon the equivalent mass of water vapour in the atmosphere (computed from temperature and humidity measurements) the absorption effect of water molecules was computed. This was the only molecular absorption mechanism considered significant. The measured value for visibility was used to estimate the effect of Mie scattering. Together with the known effect of Rayleigh scattering at that wavelength, these were combined to give the so-called total normal optical thickness of the atmosphere. Its value is for this example

$$\tau = 0.15$$

The transmittance of the atmosphere for an angle of incidence  $\theta$  of the path is given by

$$T = \exp(-\tau \sec \theta)$$

Thus for a solar zenith angle of 38° (at the time of overpass) and a nadir viewing satellite we have (see Fig. 2.1)

$$T_\theta = 0.827$$

$$T_\phi = 0.861$$

In the waveband of interest Forster notes that the solar irradiance at the earth's surface in the absence of an atmosphere is  $E_0 = 256 \text{ Wm}^{-2}$ . He further computes that the total global irradiance at the earth's surface is  $186.6 \text{ Wm}^{-2}$ . Noting from (2.4) that the term in brackets is the global irradiance this leaves the total diffuse sky irradiance as  $19.6 \text{ Wm}^{-2}$  – i.e. about 10% of the global irradiance for this example.

Based upon correction algorithms given by Turner and Spencer (1972) which account for Rayleigh and Mie scattering and atmospheric absorption Forster computes

the path radiance for this example as

$$L_p = 0.62 \text{ Wm}^{-2}\text{sr}^{-1}$$

so that (2.4) becomes

$$\begin{aligned} L_s &= R_7 0.274(186.6) + 0.62 \\ \text{i.e. } L_s &= 51.5R_7 + 0.62 \end{aligned} \quad (2.5)$$

where the subscript on  $R$  refers to the band.

For the band 7 sensors on Landsat 2 at the time of overpass it can be established in (2.3) that

$$\begin{aligned} k &= (L_{max} - L_{min})/C_{max} \\ &= (39.1 - 1.1)/63 \text{ Wm}^{-2}\text{sr}^{-1} \text{ per digital value} \\ &= 0.603 \end{aligned}$$

so that (2.3) becomes

$$L_s = 0.603 C_7 + 1.1 \quad \text{Wm}^{-2}\text{sr}^{-1}$$

which when combined with (2.5) gives

$$\begin{aligned} R_7 &= 0.0118 C_7 + 0.0094 \\ \text{or} \quad &= 1.18 C_7 + 0.94\% \end{aligned}$$

This gives a means by which the % reflectance in band 7 can be computed from the digital count value available in the digital image data. By carrying out similar computations for the other three MSS bands the absolute and differential effects of the atmosphere can be removed. For band 5 for example

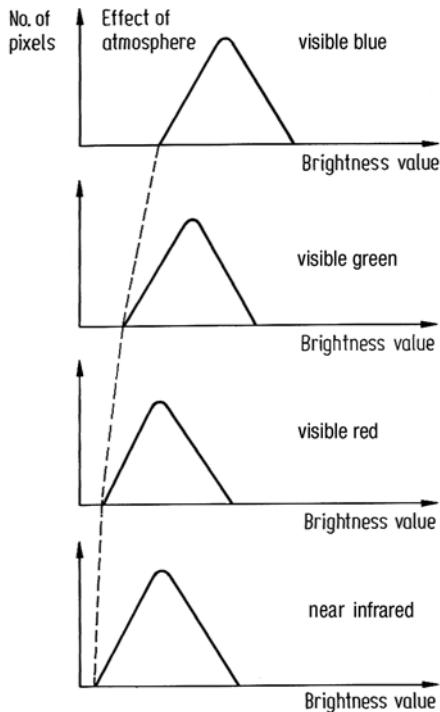
$$R_5 = 0.44 C_5 + 0.5\%$$

Note that the effects of the atmosphere (and path radiance in particular) are greater for band 5. This is because of the increasing effect of scattering with decreasing wavelength. If all four MSS bands were considered the effect would be greatest in band 4 and least in band 7.

## 2.2.2

### Bulk Correction of Atmospheric Effects

Frequently, detailed correction for the scattering and absorbing effects of the atmosphere is not required and often the necessary ancillary information such as visibility and relative humidity is not readily available. In those cases, if the effect of the atmosphere is judged to be a problem in imagery, approximate correction can be carried out in the following manner. First it is assumed that each band of data for a given scene should have contained some pixels at or close to zero brightness value but that atmospheric effects, and especially path radiance, has added a constant value to each pixel in a band. Consequently if histograms are taken of each band (i.e. graphs of the number of pixels present as a function of brightness value for a given pixel) the lowest



**Fig. 2.3.** Illustration of the effect of path radiance, resulting from atmospheric scattering, on the histograms of four bands of image data at different wavelengths

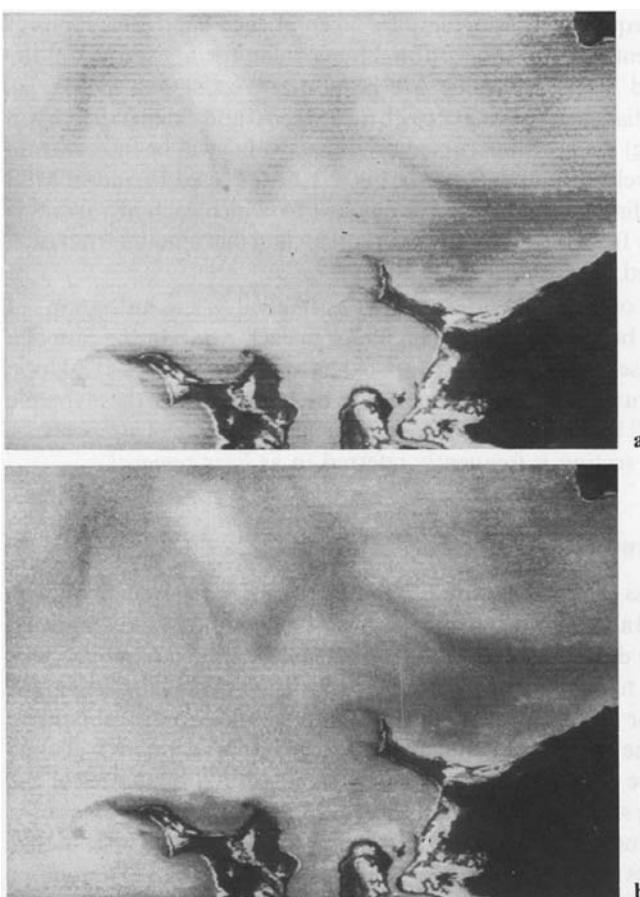
significant occupied brightness value will be non-zero as shown in Fig. 2.3. Moreover because path radiance varies as  $\lambda^{-\alpha}$  (with  $\alpha$  between 0 and 4 depending upon the extent of Mie scattering) the lowest occupied brightness value will be further from the origin for the lower wavelengths as depicted in Fig. 2.3. Correction amounts first to identifying the amount by which each histogram is “shifted” in brightness away from the origin and then subtracting that amount from each pixel brightness in that band.

It is clear that the effect of atmospheric scattering as implied in the histograms of Fig. 2.3 is to lift the overall brightness value of an image in each band. In the case of a colour composite product (see Sect. 3.2) this will appear as a whitish-bluish haze. Upon correction in the manner just described this haze will be removed and the dynamic range of image intensity will be improved. Consequently the procedure of atmospheric correction outlined in this section is frequently referred to as *haze removal*.

### 2.2.3

#### Correction of Instrumentation Errors

Errors in relative brightness such as the within-band line striping referred to in Sect. 2.1.3 and shown in Fig. 2.4a can be rectified to a great extent in the following way. First it is assumed that the detectors used for data acquisition within a band produce signals statistically similar to each other. In other words if the means and standard deviations are computed for the signals recorded by the detectors then they should be the same. This requires the assumption that detail within a band doesn't change significantly over a distance equivalent to that of one scan covered by the set of the detectors (e.g. 474 m for the six scan lines of Landsats 1,2,3 MSS). This is a reasonable assumption in terms of the mean and standard deviation of the pixel brightness, so that differences in those statistics among the detectors can be attributed



**Fig. 2.4.** **a** Landsat MSS visible green image showing severe line striping; **b** The same image after destriping by matching the mean brightnesses and standard deviations of each detector

to gain and offset mismatches as displayed in Fig. 2.2b. These mismatches can be detected by calculating pixel mean brightness and standard deviation using lines of image data known to come from a single detector. In the case of Landsat MSS this will require the data on every sixth line to be used. In a like manner five other measurements of mean brightness and standard deviation are computed as indications of the performances of the other five MSS detectors. Correction of radiometric mismatches among the detectors can then be effected by adopting one sensor as a standard and adjusting the brightness of all pixels recorded by each other detector so that their mean brightnesses and standard deviations match those of the standard detector. This can be done according to

$$y = \frac{\sigma_d}{\sigma_i}x + m_d - \frac{\sigma_d}{\sigma_i}m_i \quad (2.6)$$

where  $x$  is the old brightness of a pixel and  $y$  is its new (destriped) value;  $m_d$  and  $\sigma_d$  are the reference values of mean brightness and standard deviation and  $m_i$  and  $\sigma_i$  are the mean and standard deviation of the detector under consideration. Alternatively an independent reference mean brightness and standard deviation can be used. This can allow a degree of contrast enhancement to be produced during radiometric correction.

The method described is frequently referred to as destriping. Figure 2.4 gives an example of destriping a Landsat MSS image in this manner.

The destriping effected by (2.6) is straightforward, but capable only of matching detector responses on the basis of means and standard deviations. A more complete destriping procedure should result if the histograms of the remaining detectors are matched fully to that of the reference detector using the methods of Sect. 4.5. This approach has been used by Weinreb et al. (1989) for destriping weather satellite imagery.

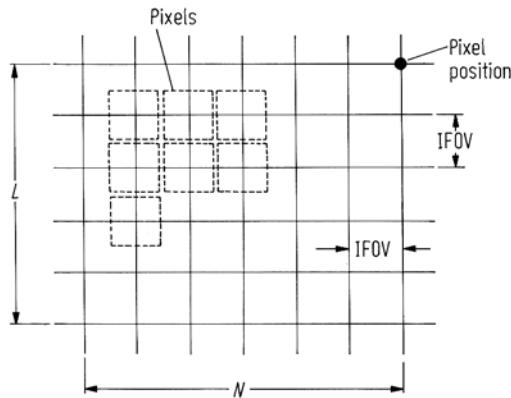
## 2.3 Sources of Geometric Distortion

There are potentially many more sources of geometric distortion of image data than radiometric distortion and their effects are more severe. They can be related to a number of factors, including

- (i) the rotation of the earth during image acquisition,
- (ii) the finite scan rate of some sensors,
- (iii) the wide field of view of some sensors,
- (iv) the curvature of the earth,
- (v) sensor non-idealities,
- (vi) variations in platform altitude, attitude and velocity, and
- (vii) panoramic effects related to the imaging geometry.

It is the purpose of this section to discuss the nature of the distortions that arise from these effects; Sect. 2.4 discusses means by which the distortions can be compensated.

To appreciate why geometric distortion occurs, in some cases it is necessary to envisage how an image is formed from sequential lines of image data. If one imagines



**Fig. 2.5.** Display grid commonly used to build up an image from the digital data stream of pixels generated by a sensor

that a particular sensor records  $L$  lines of  $N$  pixels each then it would be natural to form the image by laying the  $L$  lines down successively one under the other. If the IFOV of the sensor has an aspect ratio of unity – i.e. the pixels are the same size along and across the scan – then this is the same as arranging the pixels for display on a square grid, such as that shown in Fig. 2.5. The grid intersections are the pixel positions and the spacing between those grid points is equal to the sensor's IFOV.

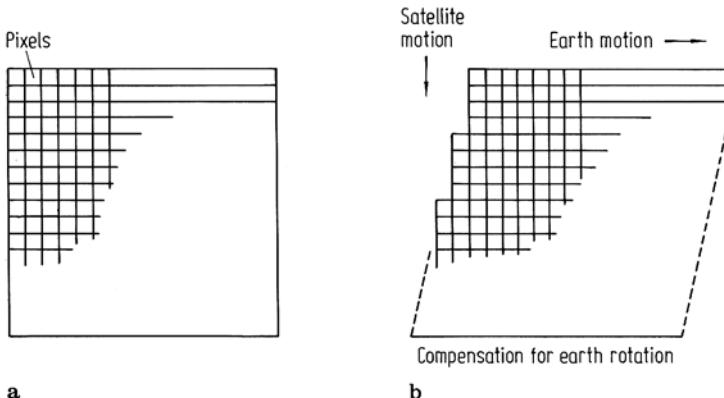
### 2.3.1 Earth Rotation Effects

Line scan sensors (Fig. 1.6) such as the Landsat TM, and the MODIS on Aqua take a finite time to acquire a frame of image data. The same is true of push broom scanners such as the SPOT HRV (Fig. 1.7). During the frame acquisition time the earth rotates from west to east so that a point imaged at the end of the frame would have been further to the west when recording started. Therefore if the lines of image data recorded were arranged for display in the manner of Fig. 2.5 the later lines would be erroneously displaced to the east in terms of the terrain they represent. Instead, to give the pixels their correct positions relative to the ground it is necessary to offset the bottom of the image to the west by the amount of movement of the ground during image acquisition, with all intervening lines displaced proportionately as depicted in Fig. 2.6. The amount by which the image has to be skewed to the west at the end of the frame depends upon the relative velocities of the satellite and earth and the length of the image frame recorded. An example is presented here for Landsat 7.

The angular velocity of the satellite is  $\omega_0 = 1.059 \text{ mrad s}^{-1}$  so that a nominal  $L = 185 \text{ km}$  frame on the ground is scanned in

$$t_s = L/(r_e \omega_0) = 27.4 \text{ s}$$

where  $r_e$  is the radius of the earth (6.37816 Mm).



**Fig. 2.6.** The effect of earth rotation on scanner imagery. **a** Image formed according to Fig. 2.5 in which lines are arranged on a square grid; **b** Offset of successive lines to the west to correct for the rotation of the earth's surface during the frame acquisition time

The surface velocity of the earth is given by

$$v_e = \omega_e r_e \cos \lambda$$

where  $\lambda$  is latitude and  $\omega_e$  is the earth rotational velocity of  $72.72 \mu\text{rad s}^{-1}$ . At Sydney, Australia  $\lambda = 33.8^\circ$  so that

$$v_e = 385.4 \text{ ms}^{-1}$$

During the frame acquisition time the surface of the earth moves to the east by

$$\Delta x_e = v_e t_s = 10.55 \text{ km at } 33.8^\circ \text{S latitude}$$

This represents 6% of the frame size. Since the satellite does not pass directly north-south this movement has to be corrected by the inclination angle. At Sydney this is approximately  $11^\circ$  so that the effective sideways movement of the earth is

$$\Delta_x = \Delta x_e \cos 11^\circ = 10.34 \text{ km}$$

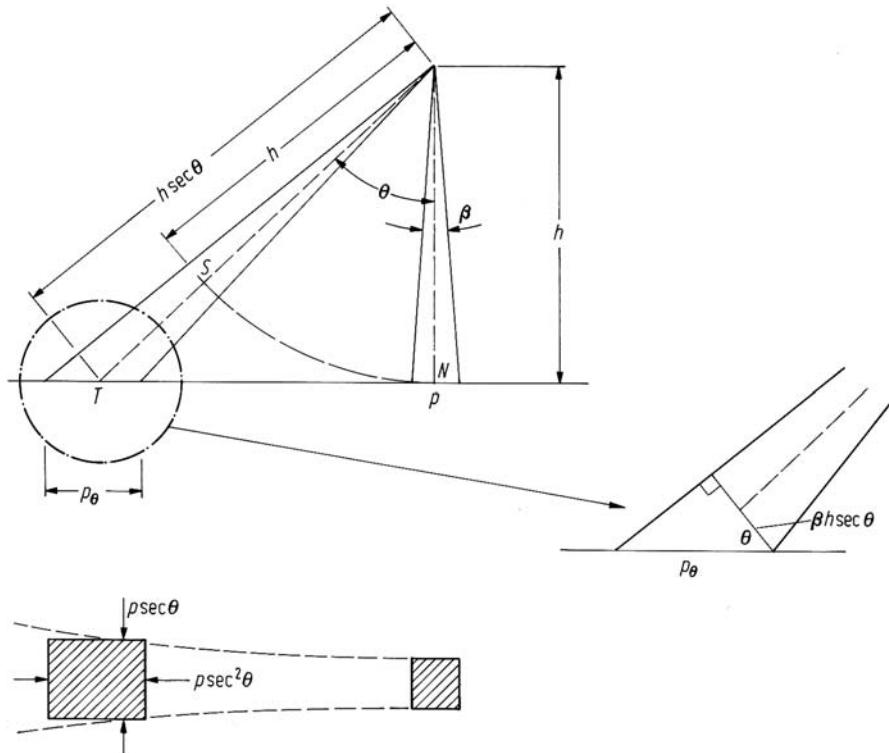
Consequently if steps are not taken to correct an image from Landsat 7 for the effect of earth rotation then the image will contain about a 6% skew distortion to the east.

### 2.3.2

#### Panoramic Distortion

For scanners used on spacecraft and aircraft remote sensing platforms the angular IFOV is constant. As a result the effective pixel size on the ground is larger at the extremities of the scan than at nadir, as illustrated in Fig. 2.7. In particular, if the IFOV is  $\beta$  and the pixel dimension at nadir is  $p$  then its dimension in the scan direction at a scan angle of  $\theta$  as shown is

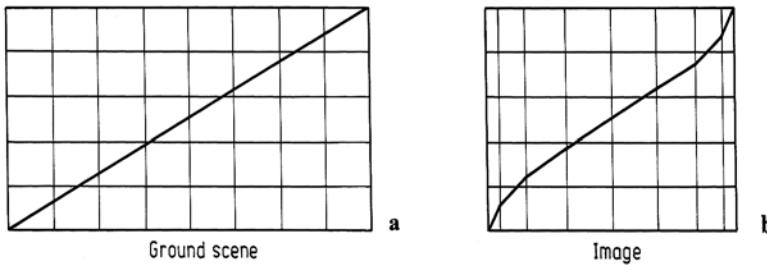
$$p_\theta = \beta h \sec^2 \theta = p \sec^2 \theta \quad (2.7a)$$



**Fig. 2.7.** Effect of scan angle on pixel size at constant angular instantaneous field of view

where  $h$  is altitude. Its dimension across the scan line is  $p \sec \theta$ . For small values of  $\theta$  these effects are negligible. For example, for Landsat 7 the largest value of  $\theta$  is approximately  $7.5^\circ$  so that  $p_\theta = 1.02 p$ . However for systems with larger fields of view, such as MODIS and aircraft scanners, the effect can be quite severe. For an aircraft scanner with  $\text{FOV} = 80^\circ$  the distortion in pixel size along the scan line is  $p_\theta = 1.70 p$  – i.e. the region on the ground measured at the extremities of the scan is 70% larger laterally than the region sensed at nadir. When the image data is arranged to form an image, as in Fig. 2.5, the pixels are all written as the same size spots on a photographic emulsion or are displayed as the same pixel size on a colour display device. Therefore the displayed pixels are equal across the scan line whereas the equivalent ground areas covered are not. This gives a compression of the image data towards its edges.

There is a second distortion introduced with wide field of view systems and that relates to pixel positions across the scan line. The scanner records pixels at constant angular increments and these are displayed on a grid of uniform centres, as in Fig. 2.5. However the spacings of the effective pixels on the ground increase with scan angle. For example if the pixels are recorded at an angular separation equal to the IFOV of the sensor then at nadir the pixels centres are spaced  $p$  apart. At a scan angle  $\theta$



**Fig. 2.8.** Illustration of the along scan line compression evident in constant angular IFOV and constant angular scan rate sensors. This leads to so-called S-bend distortion, as shown

the pixel centres will be spaced  $p \sec^2 \theta$  apart as can be ascertained from Fig. 2.7. Thus by placing the pixels on a uniform display grid the image will suffer an across track compression. Again the effect for small angular field of view systems will be negligible in terms of the relative spacings of adjacent pixels. However when the effect is compounded to determine the location of a pixel at the swath edge relative to nadir the error can be significant. This can be determined by computing the arc  $SN$  in Fig. 2.7  $S$  being the position to which the pixel at  $T$  would appear to be moved if the data is arrayed uniformly. It can be shown readily that  $SN/TN = \theta/\tan \theta$  this being the degree of across track scale distortion. In the case of Landsat 7  $(\theta/\tan \theta)_{\max} = 0.9936$ . This indicates that a pixel at the swath edge (92.5 km from the sub-nadir point) will be 314 m out of position along the scan line compared with the ground if the pixel at nadir is in its correct location.

These panoramic effects lead to an interesting distortion in the geometry of large field of view systems. To see this consider the uniform mesh shown in Fig. 2.8a. Suppose this represents a region on the ground being imaged. For simplicity the cells in the grid could be considered to be features on the ground. Because of the compression in the image data caused by displaying equal-sized pixels on a uniform grid as discussed in the foregoing, the uniform mesh will appear as shown in Fig. 2.8b. Image pixels are recorded with a constant IFOV and at a constant angular sampling rate. The number of pixels recorded therefore over the outer grid cells in the along scan direction will be smaller than over those near nadir. In the along track direction there is no variation of pixel spacing or density with scan angle as this is established by the forward motion of the platform. Rather pixels near the swath edges will contain information in common owing to the overlapping IFOV.

Linear features such as roads at an angle to the scan direction as shown in Fig. 2.8 will appear bent in the displayed image data because of the along scan compression effect. Owing to the change in shape caused, the distortion is frequently referred to as S-bend distortion and can be a common problem with aircraft line scanners. Clearly, not only linear features are affected; rather the whole image detail near the swath edges is distorted in this manner.

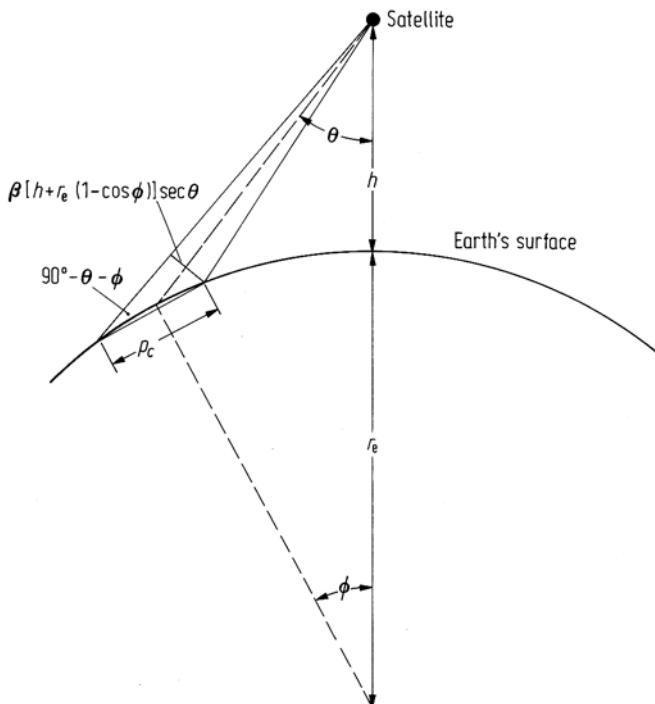
### 2.3.3

#### Earth Curvature

Aircraft scanning systems, because of their low altitude (and thus the small absolute swath width of their image data), are not affected by earth curvature. Neither are space systems such as Landsat and SPOT, again because of the narrowness of their swaths. However wide swath width spaceborne imaging systems are affected. For MODIS with a swath width of 2330 km and an altitude of 705 km it can be shown that the deviation of the earth's surface from a plane amounts to less than 1% over the swath, which seems insignificant. However it is the inclination of the earth's surface over the swath that causes the greater effect. At the edges of the swath the area of the earth's surface viewed at a given angular IFOV is larger than if the curvature of the earth is ignored. The increase in pixel size can be computed by reference to the geometry of Fig. 2.9. The pixel dimension in the across track direction normal to the direction of the sensor is  $\beta[h + r_e(1 - \cos \phi)] \sec \theta$  as shown. The geometry of Fig. 2.9 then shows that the effective pixel size on the inclined earth's surface is

$$p_c = \beta[h + r_e(1 - \cos \phi)] \sec \theta \sec(\theta + \phi) \quad (2.7b)$$

where  $\beta h$  is the pixel size at nadir and  $\phi$  is the angle subtended at the centre of the earth. Note that this expression reduces to (2.7a) if  $\phi = 0$  – i.e. if earth curvature



**Fig. 2.9.** Effect of earth curvature on the size of a pixel in the scan direction (across track)

is considered negligible. Using the NOAA satellite as an example  $\theta = 54^\circ$  at the edge of the swath and  $\phi = 12^\circ$ . This shows that the effective pixel size in the along scan direction is 2.89 times larger than that at nadir when earth curvature is ignored, but is 4.94 times that at nadir when the effect of earth curvature is included. This demonstrates that earth curvature introduces a significant additional compressive distortion in the image data acquired by satellites such as NOAA when an image is constructed on a uniform grid such as that in Fig. 2.5. The effect of earth curvature in the along track direction is negligible.

### 2.3.4

#### **Scan Time Skew**

Mechanical line scanners such as the Landsat MSS and TM require a finite time to scan across the swath. During this time the satellite is moving forward leading to a skewing in the along track direction. As an illustration of the magnitude of the effect, the time required to record one MSS scan line of data is 33 ms. During this time the satellite travels forward by 213 m at its equivalent ground velocity of  $6.467 \text{ km s}^{-1}$ . As a result the end of the scan line is advanced by this amount compared with its start.

### 2.3.5

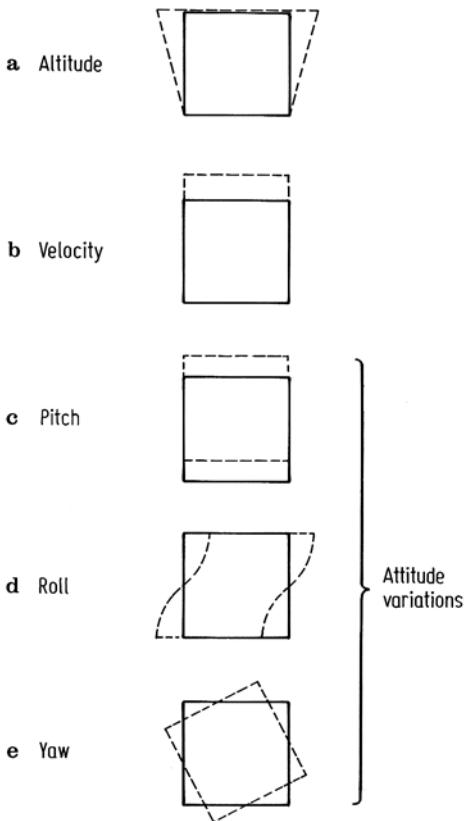
#### **Variations in Platform Altitude, Velocity and Attitude**

Variations in the elevation or altitude of a remote sensing platform lead to a scale change at constant angular IFOV and field of view; the effect is illustrated in Fig. 2.10a for an increase in altitude with travel at a rate that is slow compared with a frame acquisition time. Similarly, if the platform forward velocity changes, a scale change occurs in the along track direction. This is depicted in Fig. 2.10b again for a change that occurs slowly. For a satellite platform, orbit velocity variations can result from orbit eccentricity and the non-sphericity of the earth.

Platform attitude changes can be resolved into yaw, pitch and roll during forward travel. These lead to image rotation, along track and across track displacement as noted in Fig. 2.10 c–e.

While these variations can be described mathematically, at least in principle, a knowledge of the platform ephemeris is required to enable their magnitudes to be computed. In the case of satellite platforms ephemeris information is often telemetered to ground receiving stations. This can be used to apply corrections before the data is distributed.

Attitude variations in aircraft remote sensing systems can potentially be quite significant owing to the effects of atmospheric turbulence. These can occur over a short time, leading to localised distortions in aircraft scanner images. Frequently aircraft roll is compensated for in the data stream. This is made possible by having a data window that defines the swath width; this is made smaller than the complete scan of data over the sensor field of view. A gyro mounted on the sensor is then used



**Fig. 2.10.** Effect of platform position and attitude errors on the region of earth being imaged, when those errors occur slowly compared with image acquisition

to move the position of the data window along the total scan line as the aircraft rolls. Pitch and yaw are generally not corrected unless the sensor is mounted on a three axis stabilized platform.

A comprehensive discussion of the nature and effects of aircraft scanner distortion is given by Silva (1978).

### 2.3.6 Aspect Ratio Distortion

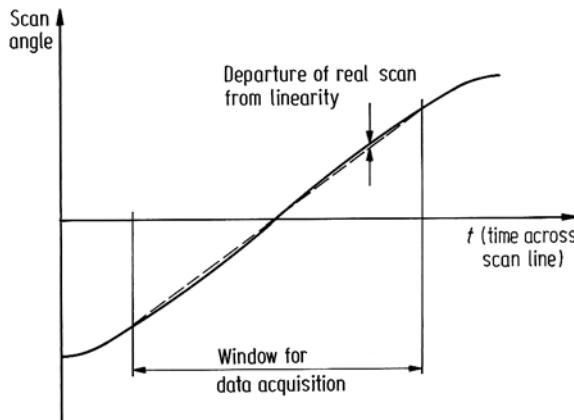
The aspect ratio of an image (that is, its scale vertically compared with its scale horizontally) can be distorted by mechanisms that lead to overlapping IFOV's. The most notable example of this occurs with the Landsat multispectral scanner. As discussed in Sect. A.1.2 samples are taken across a scan line too quickly compared with the IFOV. This leads to pixels having 56 metre centres but sampled with an IFOV of 79 m. Consequently the effective pixel size is 79 m × 56 m and thus

is not square. As a result if the pixels recorded by the multispectral scanner are displayed on the square grid of Fig. 2.5 the image will be too wide for its height when related to the corresponding region on the ground. The magnitude of the distortion is  $79/56 = 1.411$  so that this is quite a severe error and must be corrected for most applications.

A similar distortion can occur with aircraft scanners if the velocity of the aircraft is not matched to the scanning rate of the sensor. Either underscanning or overscanning can occur leading to distortion in the alongtrack scale of the image.

### 2.3.7 Sensor Scan Nonlinearities

Line scanners that make use of rotating mirrors, such as the NOAA AVHRR and aircraft scanners, have a scan rate across the swath that is constant, to the extent that the scan motor speed is constant. Systems that use an oscillating mirror however, such as the Landsat thematic mapper, incur some nonlinearity in scanning near the swath edges owing to the need for the mirror to slow down and change directions. This effect is depicted in Fig. 2.11. According to Anuta (1973) this can lead to a maximum displacement in pixel position compared with a perfectly linear scan of about 395 m, for example, for Landsat multispectral scanner products.



**Fig. 2.11.** Mirror displacement versus time in an oscillating mirror scanner system. Note that data acquisition does not continue to the extremes of the scan so that major nonlinearities are obviated

## 2.4

### Correction of Geometric Distortion

There are two techniques that can be used to correct the various types of geometric distortion present in digital image data. One is to model the nature and magnitude of the sources of distortion and use these models to establish correction formulae. This technique is effective when the types of distortion are well characterized, such as that caused by earth rotation. The second approach depends upon establishing mathematical relationships between the addresses of pixels in an image and the corresponding coordinates of those points on the ground (via a map). These relationships can be used to correct the image geometry irrespective of the analyst's knowledge of the source and type of distortion. This procedure will be treated first since it is the most commonly used and, as a technique, is independent of the platform used for data acquisition. Correction by mathematical modelling is discussed later. Before proceeding it should be noted that each band of image data has to be corrected. However since it can often be assumed that the bands are well registered to each other, steps taken to correct one band in an image, can be used on all remaining bands.

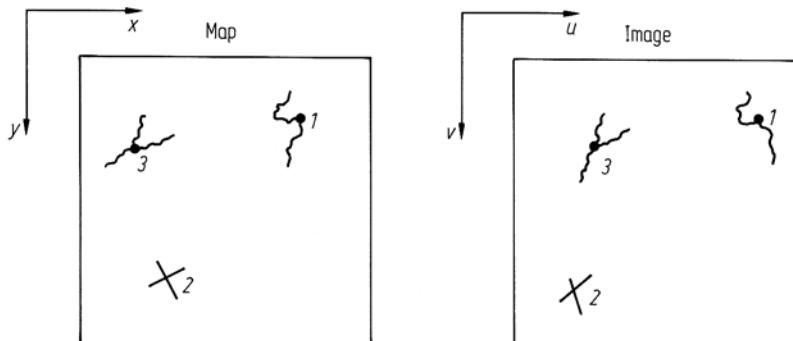
#### 2.4.1

##### Use of Mapping Polynomials for Image Correction

An assumption that is made in this procedure is that there is available a map of the region corresponding to the image, that is correct geometrically. We then define two cartesian coordinate systems as shown in Fig. 2.12. One describes the location of points in the map ( $x, y$ ) and the other coordinate system defines the location of pixels in the image ( $u, v$ ). Now suppose that the two coordinate systems can be related via a pair of mapping functions  $f$  and  $g$  so that

$$u = f(x, y) \quad (2.8a)$$

$$v = g(x, y) \quad (2.8b)$$



**Fig. 2.12.** Coordinate systems defined for the image and map, along with the specification of ground control points

If these functions are known then we could locate a point in the image knowing its position on the map. In principle, the reverse is also true. With this ability we could build up a geometrically correct version of the image in the following manner. First we define a grid over the map to act as the grid of pixel centres in the corrected image. This grid is parallel to, or indeed could in fact be, the map coordinate grid itself, described by latitudes and longitudes, UTM coordinates and so on. For simplicity we will refer to this grid as the display grid; by definition this is geometrically correct. We then move over the display grid pixel centre by pixel centre and use the mapping functions above to find the corresponding pixel in the image for each display grid position. Those pixels are then placed on the display grid. At the conclusion of the process we have a geometrically correct image built up on the display grid utilizing the original image as a source of pixels.

While the process is a straightforward one there are some practical difficulties that must be addressed. First we do not know the explicit form of the mapping functions of (2.8). Secondly, even if we did, they may not point exactly to a pixel in the image corresponding to a display grid location; instead some form of interpolation may be required.

#### **2.4.1.1 Mapping Polynomials and Ground Control Points**

Since explicit forms for the mapping functions in (2.8) are not known they are generally chosen as simple polynomials of first, second or third degree. For example, in the case of second degree (or order)

$$u = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2 \quad (2.9a)$$

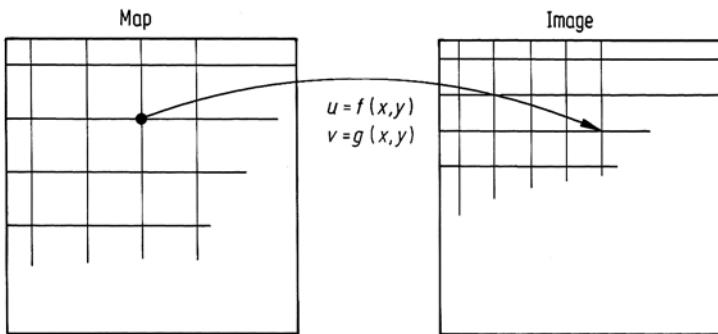
$$v = b_0 + b_1x + b_2y + b_3xy + b_4x^2 + b_5y^2 \quad (2.9b)$$

Sometimes orders higher than three are used but care must be taken to avoid the introduction of worse errors than those to be corrected, as will be noted later.

If the coefficients  $a_i$  and  $b_i$  in (2.9) were known then the mapping polynomials could be used to relate any point in the map to its corresponding point in the image as in the foregoing discussion. At present however these coefficients are unknown. Values can be estimated by identifying sets of features on the map that can also be identified on the image. These features, often referred to as *ground control points* (G.C.P's), are well-defined and spatially small and could be road intersections, airport runway intersections, bends in rivers, prominent coastline features and the like. Enough of these are chosen (as pairs – on the map and image as depicted in Fig. 2.12) so that the polynomial coefficients can be estimated by substitution into the mapping polynomials to yield sets of equations in those unknowns. Equations (2.9) show that the minimum number required for second order polynomial mapping is six. Likewise a minimum of three is required for first order mapping and ten for third order mapping. In practice however significantly more than these are chosen and the coefficients are evaluated using least squares estimation. In this manner any control points that contain significant positional errors either on the map or in the image will not have an undue influence on the polynomial coefficients.

### 2.4.1.2 Resampling

Having determined the mapping polynomials explicitly by use of the ground control points the next step is to find points in the image corresponding to each location in the pixel grid previously defined over the map. The spacing of that grid is chosen according to the pixel size required in the corrected image and need not be the same as that in the original geometrically distorted version. For the moment suppose that the points located in the image correspond exactly to image pixel centres. Then those pixels are simply transferred to the appropriate locations on the display grid to build up the rectified image. This is the case in Fig. 2.13.



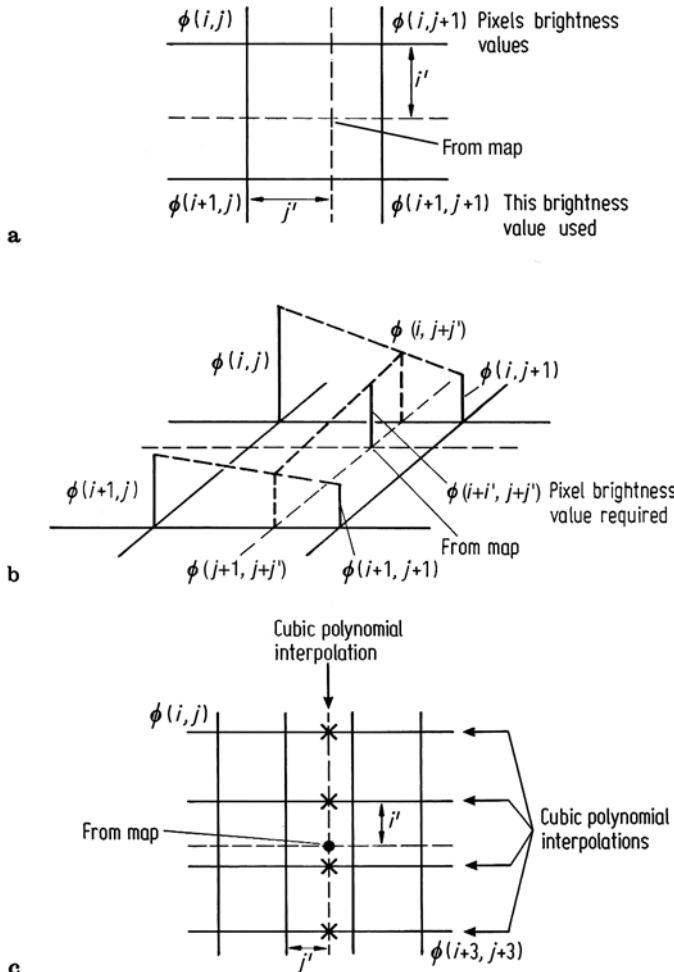
**Fig. 2.13.** Use of the mapping polynomials to locate points in the image corresponding to display grid positions

### 2.4.1.3 Interpolation

As is to be expected, grid centres from the map-registered pixel grid will not usually project to exact pixel centre locations in the image, as shown in Fig. 2.13, and some decision has to be made therefore about what pixel brightness value should be chosen for placement on the new grid. Three techniques can be used for this purpose.

*Nearest neighbour resampling* simply chooses the actual pixel that has its centre nearest the point located in the image, as depicted in Fig. 2.14a. This pixel is then transferred to the corresponding display grid location. This is the preferred technique if the new image is to be classified since it then consists of the original pixel brightnesses, simply rearranged in position to give a correct image geometry.

*Bilinear interpolation* uses three linear interpolations over the four pixels that surround the point found in the image corresponding to a given display grid position. The process is illustrated in Fig. 2.14b. Two linear interpolations are performed along the scan lines to find the interpolants  $\phi(i, j + j')$  and  $\phi(i + 1, j + j')$  as shown.



**Fig. 2.14.** Determining a display grid pixel brightness by **a** nearest neighbour resampling, **b** bilinear interpolation and **c** cubic convolution interpolation;  $i, j$  etc. are discrete values of  $u$  and  $v$

These are given by

$$\begin{aligned}\phi(i, j + j') &= j'\phi(i, j + 1) + (1 - j')\phi(i, j) \\ \phi(i + 1, j + j') &= j'\phi(i + 1, j + 1) + (1 - j')\phi(i + 1, j)\end{aligned}$$

where  $\phi$  is pixel brightness and  $(i + i', j + j')$  is the position at which an interpolated value for brightness is required. The position is measured with respect to  $(i, j)$  and assumes a grid spacing of unity in both directions. The final step is to interpolate linearly over  $\phi(i, j + j')$  and  $\phi(i + 1, j + j')$  to give

$$\begin{aligned}\phi(i + i', j + j') = & (1 - i')(j'\phi(i, j + 1) + (1 - j')\phi(i, j)) \\ & + i'\{j'\phi(i + 1, j + 1) + (1 - j')\phi(i + 1, j)\}\end{aligned}\quad (2.10)$$

*Cubic convolution interpolation* uses the surrounding sixteen pixels. Cubic polynomials are fitted along the four lines of four pixels surrounding the point in the image, as depicted in Fig. 2.14c to form four interpolants. A fifth cubic polynomial is then fitted through these to synthesise a brightness value for the corresponding location in the display grid.

The actual form of polynomial that is used for the interpolation is derived from considerations in sampling theory and issues concerned with constructing a continuous function (i.e. interpolating) from a set of samples. These are beyond the scope of this treatment but can be appreciated using the material presented in Chap. 7. An excellent treatment of the problem has been given by Shlien (1979), who discusses several possible cubic polynomials that could be used for the interpolation process and who also demonstrates that the interpolation is a convolution operation. Based on the choice of a suitable polynomial (attributable to Simon (1975)) the algorithm that is used to perform cubic convolution interpolation is (Moik, 1980):

$$\begin{aligned}\phi(i, j + 1 + j') = & j'\{j'[\phi(i, j + 3) - \phi(i, j + 2) + \phi(i, j + 1) - \phi(i, j)] \\ & + [\phi(i, j + 2) - \phi(i, j + 3) - 2\phi(i, j + 1) + 2\phi(i, j)] \\ & + [\phi(i, j + 2) - \phi(i, j)]\} \\ & + \phi(i, j + 1)\end{aligned}\quad (2.11a)$$

This expression is evaluated for each of the four lines of four pixels depicted in Fig. 2.14c to yield the four interpolants  $\phi(i, j + 1 + j')$ ,  $\phi(i + 1, j + 1 + j')$ ,  $\phi(i + 2, j + 1 + j')$ ,  $\phi(i + 3, j + 1 + j')$ . These are then interpolated vertically according to

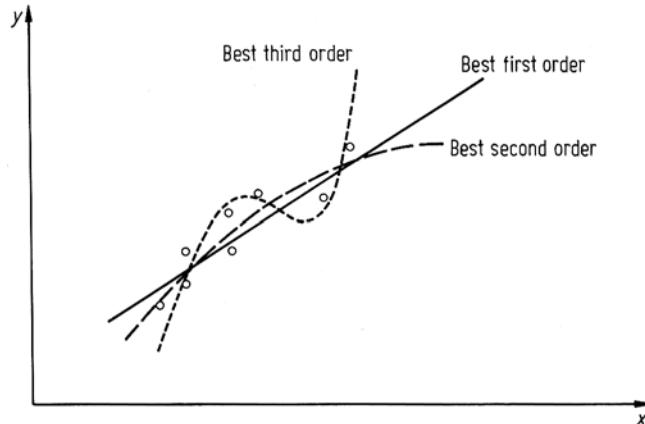
$$\begin{aligned}\phi(i + 1 + i', j + 1 + j') = & i'\{i'[\phi(i + 3, j + 1 + j') - \phi(i + 2, j + 1 + j') \\ & + \phi(i + 1, j + 1 + j') - \phi(i, j + 1 + j')] \\ & + [\phi(i + 2, j + 1 + j') - \phi(i + 3, j + 1 + j')] \\ & - 2\phi(i + 1, j + 1 + j') + 2\phi(i, j + 1 + j')]\} \\ & + [\phi(i + 2, j + 1 + j') - \phi(i, j + 1 + j')]\} \\ & + \phi(i + 1, j + 1 + j')\end{aligned}\quad (2.11b)$$

Cubic convolution interpolation, or resampling, yields an image product that is generally smooth in appearance and is often used if the final product is to be treated by photointerpretation. However since it gives pixels on the display grid, with brightnesses that are interpolated from the original data, it is not recommended if classification is to follow since the new brightness values may be slightly different to the actual radiance values detected by the satellite sensors.

### 2.4.1.4

#### Choice of Control Points

Enough well defined control point pairs must be chosen in rectifying an image to ensure that accurate mapping polynomials are generated. However care must also be given to the locations of the points. A general rule is that there should be a distribution of control points around the edges of the image to be corrected with a scattering of points over the body of the image. This is necessary to ensure that the mapping polynomials are well-behaved over the image. This concept can be illustrated by considering an example from curve fitting. While the nature of the problem is different the undesirable effects that can be generated are similar. In Fig. 2.15 is illustrated a set of data points in a graph through which first order (linear), second order and third order curves are depicted. Note that as the order is higher the curves pass closer to the points. However if it is presumed that the data would have continued for larger values of  $x$  with much the same trend as apparent in the points plotted then clearly the linear fit will extrapolate moderately acceptably. In contrast the cubic curve can deviate markedly from the trend when used as an extrapolator. This is essentially true in geometric correction of image data: while the higher order polynomials will be accurate in the vicinity of the control points themselves, they can lead to significant errors, and thus image distortions, for regions of images outside the range of the control points. This is illustrated in the example of Sect. 2.5.4.



**Fig. 2.15.** Illustration from curve fitting to reinforce the potentially poor behaviour of high order mathematical functions when used to extrapolate

### 2.4.1.5

#### Example of Registration to a Map Grid

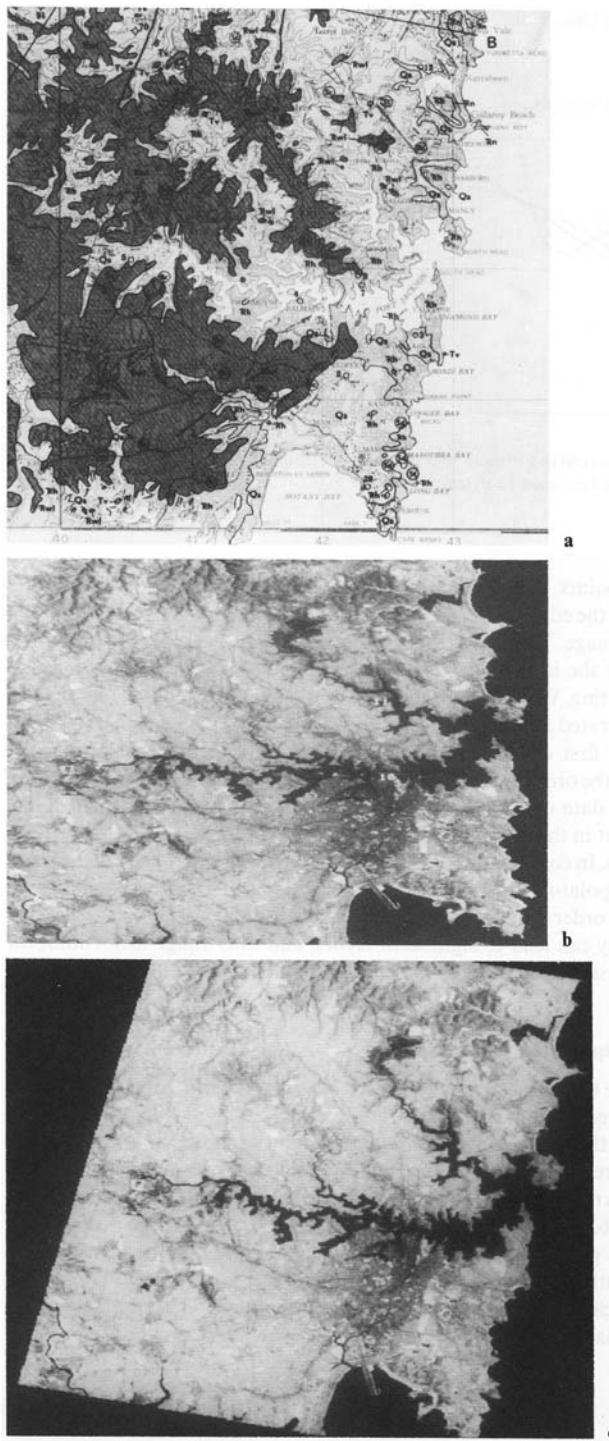
To illustrate the above techniques a small segment of a Landsat MSS image of Sydney, Australia was registered to a map of the region.

**Table 2.1.** Control points used in image to map registration example

GCP no.	Image pixel	Image line	Map easting actual	Map easting est.	Map easting residual	Map northing actual	Map northing est.	Map northing residual
1	1909.	1473.	432279.	432230.1	49.4	836471.	836410.1	60.7
2	1950.	1625.	431288.	431418.0	-130.1	822844.	822901.4	-56.9
3	1951.	1747.	428981.	428867.9	112.6	812515.	812418.2	96.8
4	1959.	1851.	427164.	427196.9	-33.2	803313.	803359.4	-46.7
5	1797.	1847.	417151.	417170.3	-18.9	805816.	805759.3	57.1
6	1496.	1862.	397860.	397871.6	-11.2	808128.	808187.2	-59.6
7	1555.	1705.	404964.	404925.8	38.6	821084.	820962.6	121.6
8	1599.	1548.	411149.	411138.5	10.5	833796.	833857.3	-61.1
9	1675.	1584.	415057.	415129.0	-72.4	829871.	829851.1	19.8
10	1829.	1713.	422019.	421986.6	32.7	816836.	816884.5	-48.1
11	1823.	1625.	423530.	423507.8	22.0	824422.	824504.8	-83.2
Standard error in easting = 55.92 m								
Standard error in northing = 63.06 m								

It is important that the map has a scale not too different from the scale at which the image data is considered useful. Otherwise the control point pairs may be difficult to establish. In this case a map at 1 : 250,000 scale was used. The relevant segment is shown reproduced in Fig. 2.16, along with the portion of image to be registered. Comparison of the two reveals the geometric distortion of the image. Eleven control points were chosen for the registration, with the coordinates shown in Table 2.1. Their UTM map coordinates were specified in this exercise by placing the map on a digitizing table, although they could have been read from the map and entered manually. The former method is substantially more convenient and often more accurate if the digitizing table facility is available.

Using the set of control points, second degree mapping polynomials were generated. To test the effectiveness of these in transferring pixels from the raw image grid to the map display grid, the software system that was used in the exercise (Dipix Systems Ltd R-STREAM) *computes* the UTM coordinates of the control points from their pixel coordinates in the image. These are compared with the actual UTM coordinates and the differences (residuals) calculated in both directions. A root mean square of all the residuals is then computed in both directions (easting and northing) as shown in Table 2.1, giving an overall impression of the accuracy of the mapping process. In this case the set of control points is seen to lead to an average positional error of 56 m in easting and 63 m in northing, which is smaller than a pixel size in equivalent ground metres and thus would be considered acceptable. At this stage the table can be inspected to see if any individual control point has residuals that are unacceptably high. It could be assumed that this is a result of poor placement; if so it could be re-entered and the polynomial recalculated. If changing that control point leaves the residuals unchanged it may be that there is local distortion in that particular region of the image. A choice has to be made then as to whether the control point should be used to give a degree of correction there, that might also influence



**Fig. 2.16.** **a** Map and **b** Landsat MSS image segment to be registered. The result obtained from second order mapping polynomials is shown in **c**

the remainder of the image, or whether it should be removed and leave that region in error.

In this example cubic convolution resampling was used in producing an image on a 50 m grid by means of the pair of second order mapping polynomials. The result is shown in Fig. 2.16.

## 2.4.2 Mathematical Modelling

If a particular distortion in image geometry can be represented mathematically then the mapping functions in (2.8) can be specified explicitly. This obviates the need to choose arbitrary polynomials as in (2.9) and to use control points to determine the polynomial coefficients. In this section some of the more common distortions are treated from this point of view. However rather than commence with expressions that relate image coordinates ( $u, v$ ) to map coordinates ( $x, y$ ) it is probably simpler conceptually to start the other way around, i.e. to model what the true (map) positions of pixels should be given their positions in an image. This expression can then be inverted if required to allow the image to be resampled on to the map grid.

### 2.4.2.1 Aspect Ratio Correction

The easiest source of distortion to model is that caused by the 56 m equivalent ground spacing of the 79 m  $\times$  79 m equivalent pixels in the Landsat multispectral scanner. As noted in Sect. 2.3.6 this leads to an image that is too wide for its height by a factor of  $79/56 = 1.411$ . Consequently to produce a geometrically correct image either the vertical dimension has to be expanded by this amount or the horizontal dimension must be compressed. We will consider the former. This requires that the pixel axis horizontally be left unchanged (i.e.  $x = u$ ), but that the axis vertically be scaled (i.e.  $y = 1.411 v$ ). This can be expressed conveniently in matrix notation as

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1.411 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \quad (2.12)$$

One way of implementing this correction would be to add extra lines of pixel data to expand the vertical scale. This could be done by duplicating four lines in every ten. Alternatively, and more precisely, (2.12) can be inverted to give

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 0.709 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.13)$$

Thus, as with the techniques of the previous section, a display grid is defined over the map (with coordinates ( $x, y$ )) and (2.13) is used to find the corresponding location in the image ( $u, v$ ). The interpolation techniques of Sect. 2.4.1.3 are then used to generate brightness values for the display grid pixels.

### 2.4.2.2

#### Earth Rotation Skew Correction

To correct for the effect of earth rotation it is necessary to implement a shift of pixels to the left that is dependent upon the particular line of pixels, measured with respect to the top of the image. Their line addresses as such ( $v$ ) are not affected. Using the results of Sect. 2.3.1, these corrections are implemented by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

with  $\alpha = -0.056$  for Sydney, Australia. Again this can be implemented in an approximate sense by making one pixel shift to the left every 17 lines of image data measured down from the top, or alternatively the expression can be inverted to give

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & -\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0.056 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.14)$$

which again is used with interpolation procedures from Sect. 2.4.1.3 to generate display grid pixels.

### 2.4.2.3

#### Image Orientation to North-South

Although not strictly a geometric distortion it is an inconvenience to have an image that is corrected for most major effects but is not oriented vertically in a north-south direction. It will be recalled for example that the Landsat orbits in particular are inclined to the north-south line by about  $9^\circ$ . (This of course is different with different latitudes). To rotate an image by an angle  $\zeta$  in the counter -or anticlockwise direction (as required in the case of Landsat) it is easily shown that (Foley, Van Dam, Feiner and Hughes, 1990)

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \zeta & \sin \zeta \\ -\sin \zeta & \cos \zeta \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

so that

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \cos \zeta & -\sin \zeta \\ \sin \zeta & \cos \zeta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.15)$$

### 2.4.2.4

#### Correction of Panoramic Effects

The discussion in Sect. 2.3.2 makes note of the pixel positional error that results from scanning with a fixed IFOV at a constant angular rate. In terms of map and image coordinates, the distortion can be described by

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \tan \theta/\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

where  $\theta$  is the instantaneous scan angle, which in turn can be related to  $x$  or  $u$ , viz.  $x = h \tan \theta$ ,  $u = h\theta$ , where  $h$  is altitude. Consequently resampling can be carried out according to

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \theta \cot \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} h/x \tan^{-1}(x/h) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \quad (2.16)$$

### 2.4.2.5 Combining the Corrections

Clearly any exercise in image correction usually requires several distortions to be rectified. Using the techniques in Sect. 2.4.1 it is assumed that all sources are rectified simultaneously. When employing mathematical modelling, a correction matrix has to be devised for each source considered important, as in the preceding sub-sections, and the set of matrices combined. For example if the aspect ratio of a Landsat TM image is corrected first, followed by correction of the effect of earth rotation, then the following single linear transformation can be established for resampling.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1.411 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1.411\alpha \\ 0 & 1.411 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

which for  $\alpha = -0.056$  (at Sydney) gives

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1 & 0.056 \\ 0 & 0.709 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

## 2.5 Image Registration

### 2.5.1 Georeferencing and Geocoding

Using the correction techniques of the preceding sections an image can be registered to a map coordinate system and therefore have its pixels addressable in terms of map coordinates (eastings and northings, or latitudes and longitudes) rather than pixel and line numbers. Other spatial data types, such as geophysical measurements, image data from other sensors and the like, can be registered similarly to the map thus creating a georeferenced integrated spatial data base of the type used in a geographic information system.

Expressing image pixel addresses in terms of a map coordinate base is often referred to as geocoding.

### 2.5.2

#### **Image to Image Registration**

Many applications of remote sensing image data require two or more scenes of the same geographical region, acquired at different dates, to be processed together. Such a situation arises for example when changes are of interest, in which case registered images allow a pixel by pixel comparison to be made.

Two images can be registered to each other by registering each to a map coordinate base separately, in the manner demonstrated in the previous section. Alternatively, and particularly if georeferencing is not important, one image can be chosen as a master to which the other, known as the slave, is to be registered. Again the techniques of Sect. 2.4 are used, however the coordinates  $(x, y)$  are now the pixel coordinates in the master image rather than the map coordinates. As before  $(u, v)$  are the coordinates of the image to be registered (i.e. the slave). An advantage in image to image registration is that only one registration step is required in comparison to two if both are taken back to a map base. Furthermore an artifice known as a sequential similarity detection algorithm can be used to assist in accurate co-location of control point pairs.

### 2.5.3

#### **Control Point Localisation by Correlation**

Correlation is of value in locating the position of a control point in the master image having identified it in the slave. The sequential similarity detection algorithm (SSDA), as treated by Bernstein (1983), is of this type. Only one specific implementation is considered here to illustrate the nature of the method. Other methods are summarized by Yao (2001).

Suppose a control point has been chosen in the slave image and it is necessary to determine its counterpart in the master image. In principle a rectangular sample of pixels surrounding the control point in the slave image can be extracted as a window to be correlated with the master image. Because of the spatial properties of the pair of images near the control points a high correlation should occur when the slave window is located over its exact counterpart region in the master, and thus the master location of the control point is identified. Obviously it is not necessary to move the slave window over the complete master image since the user knows approximately where the control point should occur in the master. Consequently it is only necessary to specify a search region in the neighbourhood of the approximate location. Software systems that provide this option allow the user to choose both the size of the window of pixels from the slave image control point neighbourhood and the size of the search region in the master image over which the window of slave pixels is moved to detect an acceptable correlation.

The correlation measure used need not be sophisticated. Indeed a simple similarity check that can be used is to compute the sum of the absolute differences of the slave and master image pixel brightnesses over the window, for each possible location of the window in the search region. The location that gives the smallest absolute difference defines the control point position as that pixel at the current centre

of the window. Obviously the sensitivity of the method will be reduced if there is a large average difference in brightness between the two images – such as that owing to seasonal variations. A refinement therefore is to compute the summed absolute difference of the pixel brightnesses relative to their respective means in the search window.

Clearly the use of techniques such as these to locate control points depends upon there not being major changes of an uncorrelated nature between the scenes in the vicinity of a control point being tested. For example a vegetation flush due to rainfall in part of the search window can lead to an erroneous location. Nevertheless with a judicious choice of window size and search region, measures such as SSDA can give very effective guidance to the user, especially when available on an interactive image processing facility.

#### **2.5.4**

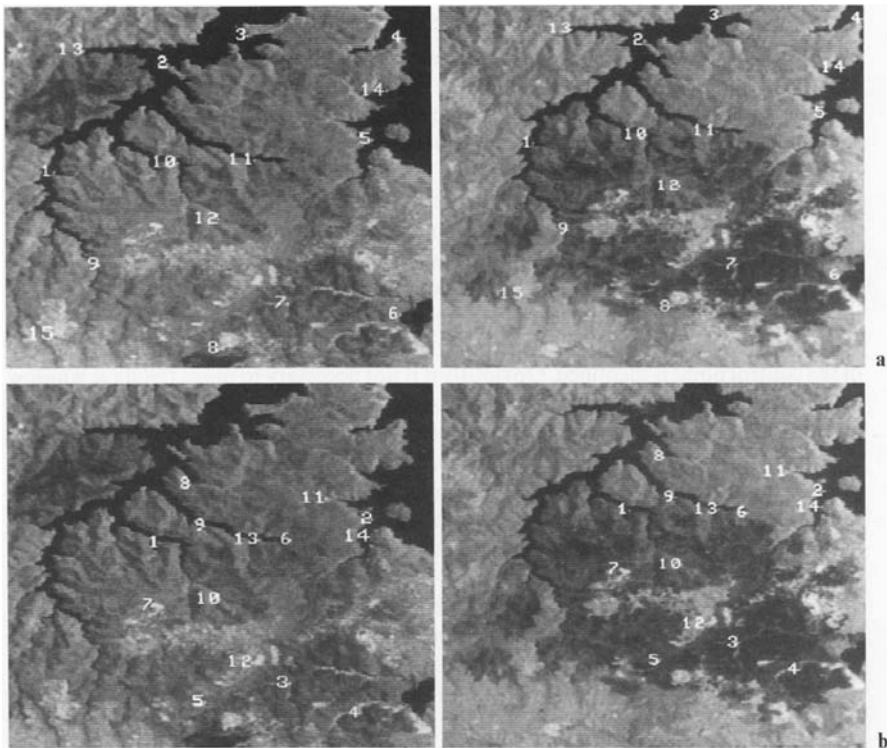
#### **Example of Image to Image Registration**

To illustrate image to image registration, but more particularly to see clearly the effect of control point distribution and the significance of the order of the mapping polynomials to be used for registration, two segments of Landsat MSS infrared image data in the northern suburbs of Sydney were chosen. One was acquired on December 29, 1979 and was used as the master. The other was acquired on December 14, 1980 and was used as the slave image. These are shown in Fig. 2.17 wherein careful inspection shows the differences in image geometry.

Two sets of control points were chosen. In one the points were distributed as nearly as possible in a uniform manner around the edge of the image segment as shown in Fig. 2.17a, with some points located across the centre of the image. This set would be expected to give a reasonable registration of the images. The second set of control points was chosen injudiciously, closely grouped around one particular region, to illustrate the resampling errors that can occur. These are shown in Fig. 2.17b. In both cases the control point pairs were co-located with the assistance of a sequential similarity detection algorithm. This worked well particularly for those control points around the coastal and river regions where the similarity between the images is unmistakable. To minimise tidal influences on the location of control points, those on water boundaries were chosen as near as possible to be on headlands, and certainly were never chosen at the ends of inlets.

For both sets of control points third degree mapping polynomials were used along with cubic convolution resampling. As expected the first set of points led to an acceptable registration of the images whereas the second set gave a good registration in the immediate neighbourhood of the points but beyond them produced gross distortion.

The adequacy of the registration process can be assessed visually if the master and resampled slave images can be superimposed in different colours. Figure 2.18a and 2.18b show the master image in red with the resampled slave image superimposed in green. Where good registration has been achieved the resultant is yellow (with the exception of regions of gross dissimilarity in pixel brightness – in this case associated



**Fig. 2.17.** Control points used in image to image registration example. **a** Good distribution; **b** Poor distribution

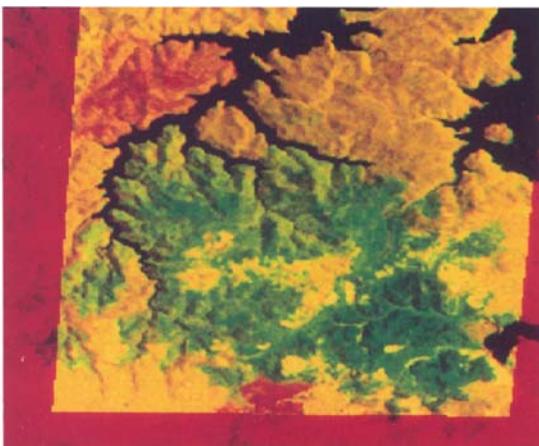
with fire burns). However misregistration shows quite graphically by a red-green separation. This is particularly noticeable in Fig. 2.18b where the poor extrapolation obtained with third order mapping is demonstrated.

The exercise using the poor set of control points (Fig. 2.17b) was repeated. However this time only first order mapping polynomials were used. While these obviously will not remove non-linear differences between the images and will give poorer matches at the control points themselves, they are well behaved in extrapolation beyond the vicinity of the control points and lead to an acceptable registration as shown in Fig. 2.18c.

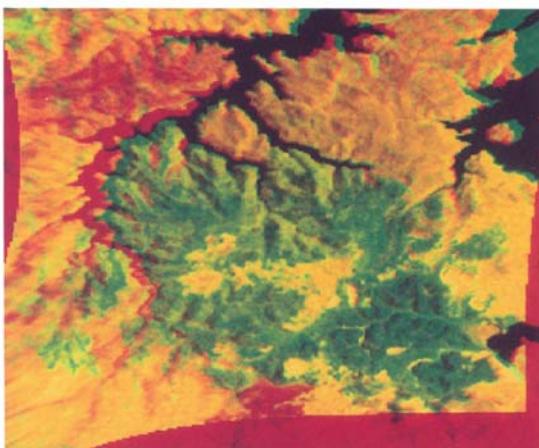
## 2.6

### Miscellaneous Image Geometry Operations

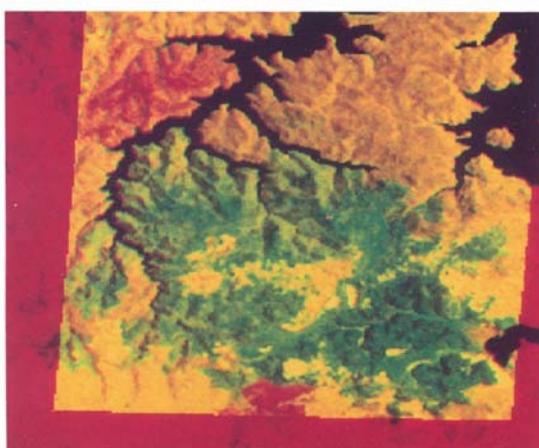
While the techniques of Sects. 2.4 and 2.5 have been devised for treating errors in image geometry and for registering sets of images, and images to maps, they can be exploited also for performing intentional changes to image geometry. Image rotation and scale changing are chosen here as illustrations.



a



b



c

**Fig. 2.18. a** Registration of 1980 image (green) with 1979 image (red) using the control points of Fig. 2.17a, with third order mapping polynomials; **b** Third order mapping of 1980 image (green) to 1979 image (red) using the control points of Fig. 2.17b; **c** As for **b** but using first order mapping polynomials

### 2.6.1

#### Image Rotation

Rotation of image data by an angle about the pixel grid can be useful for a number of applications. Most often it is used to align the pixel grid, and thus the image, to a north-south orientation as treated in Sect. 2.4.2.3. However the transformation in (2.15) is perfectly general and can be used to rotate any image in an anticlockwise sense by any specified angle  $\zeta$ .

### 2.6.2

#### Scale Changing and Zooming

The scales of an image in both the vertical and horizontal directions can be altered by the transformation.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$

where  $a$  and  $b$  are the desired scaling factors. To resample the scaled image onto the display grid we use the inverse operation, as before, to locate pixel positions in the original image corresponding to each display grid position, viz

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 1/a & 0 \\ 0 & 1/b \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}.$$

Again interpolation is used to establish actual pixel brightnesses to use, since  $u, v$  will not normally fall on exact grid locations.

Frequently  $a = b$  so that the image is simply magnified (although different magnification factors could be used in each direction if desired). This is often called zooming, particularly if the process is implemented in an image display system. If the nearest neighbour interpolation procedure is used in the resampling process the zoom implemented is said to occur by pixel replication and the image will look progressively blocky for larger zoom factors. If cubic convolution interpolation is used there will be a change in magnification but the image will not take on the blocky appearance. Often this process is called interpolative zoom. Both pixel replication zoom and interpolative zoom can also be implemented in hardware to allow the process to be performed in real time.

## References for Chapter 2

Good discussions on the effect of the atmosphere on image data in the visible and infrared wavelength ranges will be found in Slater (1980) and Forster (1984). Forster gives a detailed set of calculations to illustrate how correction procedures for compensating radiometric distortion caused by the atmosphere are applied. Definitions and derivations of radiometric quantities are covered comprehensively by Slater.

Extensive treatments of geometric distortion and means for geometric correction are covered by Anuta (1973), Billingsley (1983) and Bernstein (1983). Discussions of resampling

interpolation techniques in particular, and the optimum distribution of control points will be found in Shlien (1979) and Ortí (1981).

An interesting account of geometrical transformations in general, but especially as related to computer graphics, is found in Foley et al. (1990).

- P.E. Anuta, 1973: Geometric Correction of ERTS-1 Digital MSS Data. Information Note 103073, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.
- R. Bernstein, 1983: Image Geometry and Rectification. In R.N. Colwell (Ed.) Manual of Remote Sensing, 2e, Chapter 21, Falls Church, Va. American Society of Photogrammetry.
- F.C. Billingsley, 1983: Data Processing and Reprocessing in R.N. Colwell (Ed.) Manual of Remote Sensing, 2e, Chapter 17, Falls Church, Va. American Society of Photogrammetry.
- J.D. Foley, A. Van Dam, S.K. Feiner and J.F. Hughes, 1990: Computer Graphics Principles and Practice, 2e, Philippines, Addison-Wesley.
- B.C. Forster, 1984: Derivation of Atmospheric Correction Procedures for Landsat MSS with Particular Reference to Urban Data. *Int. J. Remote Sensing*, 5, 799–817.
- J. Yao, 2001: Image Registration Based on Both Feature and Intensity Matching. Proc. IEEE Conf. on Acoustics, Speech and Signal Processing, 3, 1693–1696.
- T.G. Moik, 1980: Digital Processing of Remotely Sensed Images, Washington, NASA.
- F. Ortí, 1981: Optimal Distribution of Control Points to Minimize Landsat Image Registration Errors. *Photogrammetric Engineering and Remote Sensing*, 47, 101–110.
- S. Shlien, 1979: Geometric Correction, Registration and Resampling of Landsat Imagery. *Canadian J. Remote Sensing*, 5, 74–89.
- L.F. Silva, 1978: Radiation and Instrumentation in Remote Sensing. In P.H. Swain & S.M. Davis (Eds.) *Remote Sensing: The Quantitative Approach*, N.Y., Mc-Graw-Hill.
- K. Simon, 1975: Digital Reconstruction and Resampling for Geometric Manipulation. Proc. Symp. on Machine Processing of Remotely Sensed Data, Purdue University, June 3–5.
- P.N. Slater, 1980: *Remote Sensing: Optics and Optical System*, Reading, Mass., Addison-Wesley.
- R.E. Turner and M.M. Spencer, 1972: Atmospheric Model for Correction of Spacecraft Data. Proc. 8th Int. Symp. on Remote Sensing of the Environment, Ann Arbor, Michigan, 895–934.
- M.P. Weinreb, R. Xie, I.H. Liennesch and D.S. Crosby, 1989: Destriping GOES Images by Matching Empirical Distribution Functions. *Remote Sensing of Environment*, 29, 185–195.

## Problems

**2.1** (a) Consider a (hypothetical) region on the ground consisting of a square grid. For simplicity suppose the grid “lines” are 79 m in width and the grid spacing is 790 m. Sketch how the region would appear in Landsat multispectral scanner imagery, before any geometric correction has been applied. Include only the effect of earth rotation and the effect of 56 m horizontal spacing of the 79 m × 79 m ground resolution elements.

(b) Develop a pair of linear (first order) mapping polynomials that will correct the image data of part (a). Assume the “lines” on the ground have a brightness of 100 and the background brightness is 20. Resample onto a 50 m grid and use a nearest neighbour interpolation. You will not want to compute all the resampled pixels unless a small computer program is used for the

purpose. Instead you may wish simply to consider some significant pixels in the resampling to illustrate the accuracy of the geometric correction.

**2.2** A sample of pixels from each of three cover types present in the Landsat MSS scene of Sydney, Australia, acquired on 14 December, 1980 is given in Table 2.2a. Only the brightnesses (digital count values) in the visible red band (0.6 to 0.7  $\mu\text{m}$ ) and the second of the infrared bands (0.8 to 1.11  $\mu\text{m}$ ) are given.

For this image Forster (1984) has computed the following relations between reflectance ( $R$ ) and the digital count values ( $C$ ) measured in the image data, where the subscript 7 refers to the infrared data and the subscript 5 refers to the visible red data:

$$R_5 = 0.44 C_5 + 0.5$$

$$R_7 = 1.18 C_7 + 0.9$$

Table 2.2b shows samples of MSS digital count values for Sydney acquired on 8 June 1980. For this scene, Forster has determined

**Table 2.2.** Pixels from three cover types in wavelength bands 5 (0.6 to 0.7  $\mu\text{m}$ ) and 7 (0.8 to 1.1  $\mu\text{m}$ ) (on a scale of 0 to 255)

(a) Landsat MSS image of Sydney, 14 December 1980

Water		Vegetation		Bare	
Band 5	Band 7	Band 5	Band 7	Band 5	Band 7
20	11	60	142	74	66
23	7	53	130	103	82
21	8	63	140	98	78
21	7	52	126	111	86
22	7	34	92	84	67
19	3	38	120	76	67
17	1	38	151	72	67
20	4	38	111	98	71
24	8	31	81	99	80
19	4	50	158	108	71

(b) Landsat MSS image of Sydney, 8 June 1980

Water		Vegetation		Bare	
Band 5	Band 7	Band 5	Band 7	Band 5	Band 7
11	2	19	41	43	27
13	5	24	45	43	34
13	2	20	44	40	30
11	1	22	30	27	19
9	1	15	22	34	23
14	4	14	26	36	26
13	4	21	27	34	27
15	5	17	38	70	50
12	4	24	37	37	30
15	4	20	27	44	30

$$R_5 = 3.64 C_5 - 1.6$$

$$R_7 = 1.52 C_7 - 2.6 .$$

Compute mean values of the digital count values for each cover type in each scene and plot these (along with bars that indicate standard deviation) in a multispectral space. This has two axes; one is the pixel digital value in the infrared band and the other is the value in the visible red band.

Instead of plotting the mean and standard deviations of the digital count values, convert the data to reflectances first. Comment on the effect correction of the raw digital count values to reflectance data (in which atmospheric effects have been removed) has on the apparent spectral separation of the three cover types.

**2.3** Aircraft line scanners acquire image data using a mirror that sweeps out lines of data at right angles to the fuselage axis. In the absence of a cross wind, scanning therefore will be orthogonal to the aircraft ground track, as is the case for satellite scanning. However aircraft data is frequently recorded in the presence of a cross wind. The aircraft fuselage then maintains an angle to the ground track so that scanning is no longer orthogonal to the effective forward motion. Discuss the nature of the distortion, referred to as "crabbing", that this causes in the image data as displayed on a rectangular grid. Remember to take account also of the finite scan time across a line.

Push broom scanners, employing linear arrays of charge coupled device sensors, are also used on aircraft. What is the nature of the geometric distortion incurred with those devices in the presence of a crosswind?

It is technically feasible to construct two dimensional detector arrays for use as aircraft sensors, with which frames of images data would be recorded in a "snapshot fashion" much like the Landsat RBV. What geometric distortion would be incurred with this device as a result of a cross-wind?

**2.4** Compute the skew distortion resulting from earth rotation in the case of Landsats 7, and Spot.

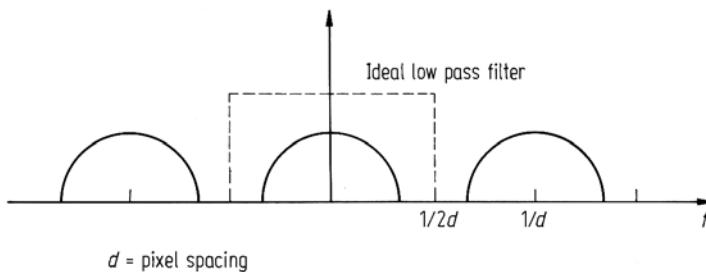
**2.5** For a particular application suppose it was necessary to apply geometric correction procedures to an image prior to *classification*. (See Chap. 3 for an overview of classification). What interpolation technique would you see use in the resampling process? Why?

**2.6** Destriping of Landsat multispectral scanner images is often performed by computing six (modulo-6 line) histograms and then either (i) matching all six to a standard histogram or (ii) choosing one of the six as a reference and matching the other five to it. Which method is to be preferred if the image is to be analysed by photointerpretation or by classification (see Chap. 3)?

**2.7** In a particular problem you have to register five Landsat images to a map. Would you register each image to the map separately, register one image to the map and then the other four images to that one, or image 1 to the map, image 2 to image 1, image 3 to image 2 etc?

**2.8** (Requires a background in digital signal processing and sampling theory).

Remote sensing digital images are simply two dimensional uniform samples of the ground scene. In particular one line of image data is a regular sequence of samples. The spatial frequency spectrum of the line data will therefore be periodic as depicted in Fig. 2.19; it is assumed here there is no aliasing. From sampling theory it is well known that the original data can be recovered by low pass filtering the spectrum, using the ideal filter as indicated in the figure. Multiplication of the spectrum by this ideal filter is equivalent to convolving the



**Fig. 2.19.** Spatial frequency spectrum of the line data

original line samples by the inverse Fourier transform of the filter function. From the theory of the Fourier transform, the inverse of the filter function is

$$s(x) = \frac{2d}{\pi} \frac{\sin x}{x}$$

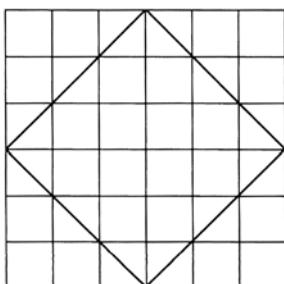
with  $x = \zeta/2d$  in which  $\zeta$  is a spatial variable along lines of data, and  $d$  is the inter-pixel spacing. This is known generally as an interpolating function. Determine some cubic polynomial approximations to this function. These could be determined from a simple Taylor series expansion or could be derived from cubic splines. For a set of examples see Shlien (1979).

**2.9** A multispectral scanner has been designed for aircraft operation. It has a field of view (FOV) of  $\pm 35^\circ$  about nadir and an instantaneous field of view (IFOV) of 2 mrad. The sensor is designed to operate at a flying height of 1000 m.

- (i) Determine the pixel size, in metres, at nadir.
- (ii) Determine the pixel size at the edge of a swath compared with that at nadir.
- (iii) Discuss the nature of the distortion in the image geometry encountered if the pixels across a scan line are displayed on uniform pixel centre.

**2.10** Determine the maximum angle of the field of view for an airborne optical sensor with a constant instantaneous field of view (IFOV), so that the pixel dimension along the scan line at the extremes is less than 1.5 times that at nadir (ignore earth curvature effect).

**2.11** Consider the panoramic along scan line distortion of an airborne optical remote sensing system with a constant instantaneous field of view (IFOV); sketch the image formed for the ground scene shown in Fig. 2.20, and explain why it appears as you have sketched it.



**Fig. 2.20.** Ground scene

# 3

## The Interpretation of Digital Image Data

### 3.1

#### Approaches to Interpretation

When image data is available in digital form, spatially quantised into pixels and radiometrically quantised into discrete brightness levels, several approaches are possible in endeavouring to extract information. One involves the use of a computer to examine each pixel in the image individually with a view to making judgement about pixels specifically based upon their attributes. This is referred to as *quantitative analysis* since pixels with like attributes are often counted to give area estimates. Means for doing this are described in Sect. 3.4. Another approach involves a human analyst/interpreter extracting information by visual inspection of an image composed from the image data. In this he or she notes generally large scale features and is often unaware of the spatial and radiometric digitisations of the data,. This is referred to as *photointerpretation* or sometimes *image interpretation*; its success depends upon the analyst exploiting effectively the spatial, spectral and temporal elements present in the composed image product. Information spatially, for example, is present in the qualities of shape, size, orientation and texture. Roads, coastlines and river systems, fracture patterns, and lineaments generally, are usually readily identified by their spatial disposition. Temporal data, such as the change in a particular object or cover type in an image from one date to another can often be used by the photointerpreter as, for example, in discriminating deciduous or ephemeral vegetation from perennial types. Spectral clues are utilised in photointerpretation based upon the analyst's fore-knowledge of, and experience with, the spectral reflectance characteristics of typical ground cover types, and how those characteristics are sampled by the sensor on the satellite or aircraft used to acquire the image data.

Those two approaches to image interpretation have their own roles and often these are complementary. Photointerpretation is aided substantially if a degree of digital image processing is applied to the image data beforehand, while quantitative analysis depends for its success on information provided at key stages by an analyst. This information very often is drawn from photointerpretation.

**Table 3.1.** A comparison of photointerpretation and quantitative analysis

Photointerpretation (by a human analyst/interpreter)	Quantitative analysis (by computer)
On a scale large relative to pixel size	At individual pixel level
Inaccurate area estimates	Accurate area estimates possible
Only limited multispectral analysis	Can perform true multispectral (multidimensional) analysis
Can assimilate only a limited number of distinct brightness levels (say 16 levels in each feature)	Can make use quantitatively of all available brightness levels in all features (e.g. 256, 1024, 4096)
Shape determination is easy	Shape determination involves complex software decisions
Spatial information is easy to use in a qualitative sense	Limited techniques available for making use of spatial data

A comparison of the attributes of photointerpretation and quantitative analysis is given in Table 3.1. From this it can be concluded that photointerpretation, involving direct human interaction and therefore high level decisions, is good for spatial assessment but poor in quantitative accuracy. Area estimates by photointerpretation, for instance, would involve planimetric measurement of regions identified visually; in this, boundary definition errors will prejudice area accuracy. By contrast, quantitative analysis, requiring little human interaction, has poor spatial ability but high quantitative accuracy. Its high accuracy comes from the ability of a computer, if required, to process every pixel in a given image and to take account of the full range of spectral, spatial and radiometric detail present. Its poor spatial properties come from the relative difficulty with which decisions about shape, size, orientation and texture can be made using standard sequential computing techniques.

In computer-based quantitative analysis the attributes of each pixel (such as the spectral bands available) are examined in order to give the pixel a label identifying it as belonging to a particular class of pixels of interest to the user. As a result, the process is often also called *classification*. We will consider that process in a little more detail shortly. In the particular case of hyperspectral data, because of the high spectral definition available, pixel identification and thus classification is possible using knowledge of the spectroscopic properties of earth surface materials. It is also a quantitative approach because identification happens at the pixel level. Although the knowledge of an expert analyst is used in performing the identification, most often the spectrum of a pixel recorded by a hyperspectral sensor is identified by comparing it against a data base of pre-recorded spectra (see Chap. 13).

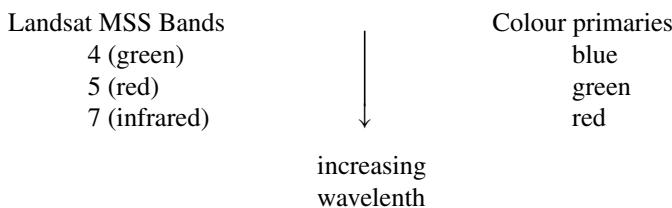
### 3.2

## Forms of Imagery for Photointerpretation

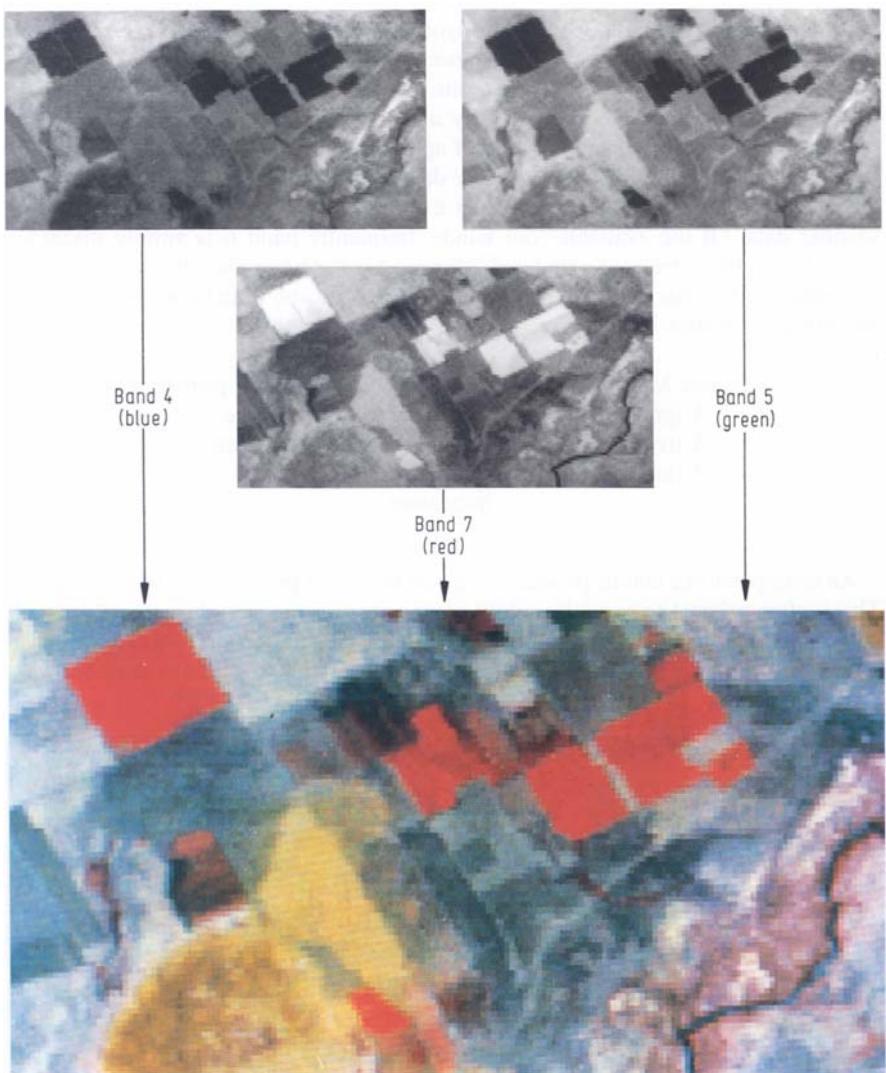
Image data can be procured either in photographic form or in digital format. The latter is more flexible since first, photographic products can be created from the digital data and secondly, the data can be processed digitally for enhancement before visual interpretation.

There are two fundamental types of display product. The first is a black and white display of each band in the image data. If produced from the raw digital data then black will correspond to a digital brightness value of 0 whereas white will correspond to the highest digital value. This is usually 63, 127, 255 or 4095 (for 6 bit, 7 bit, 8 bit and 12 bit data respectively).

The second display product is a colour composite in which selected features or bands in multispectral data are chosen to be associated with the three additive colour primaries in the display device which produces the colour product. When the data consists of more than three features a judgement has to be made as to how to discard all but three, or alternatively, a mapping has to be invented that will allow all the features to be combined suitably into the three primaries. One possible mapping is the principal components transformation developed in Chap. 6. Usually this approach is not adopted since the three new features are synthetic and the analyst is therefore not able to call upon experience of spectral reflectance characteristics. Instead a subset of original bands is chosen to form the colour composite. When the data available consists of a large number of bands (such as produced by aircraft scanners or by imaging spectrometers) only experience, and the area of application, tell which three bands should be combined into a colour product. For data with limited spectral bands however the choice is more straightforward. An example of this is Landsat multispectral scanner data. Of the available four bands, frequently band 6 is simply discarded since it is highly correlated with band 7 for most cover types and also is more highly correlated with bands 4 and 5 than is band 7. Bands 4, 5 and 7 are then associated with the colour primaries in the order of increasing wavelength:



An example of the colour product obtained by such a procedure is seen in Fig. 3.1. This is often referred to as a false colour composite or sometimes in the past, by association with colour infrared film, a colour infrared image. In this, vegetation shows as variations in red (owing to the high infrared response associated with vegetation), soils show as blue, green and sometimes yellow and water as black



**Fig. 3.1.** Formation of a Landsat multispectral scanner false colour composite by displaying the infrared band as red, the visible red band as green and the visible green band as blue

or deep blue. These colour associations are easily determined by reference to the spectral reflectance characteristics of earth surface cover types in Sect. 1.1; it is important also to take notice of the spectral imbalance created by computer enhancement of the brightness range in each wavelength band as discussed in Sect. 3.3 following.

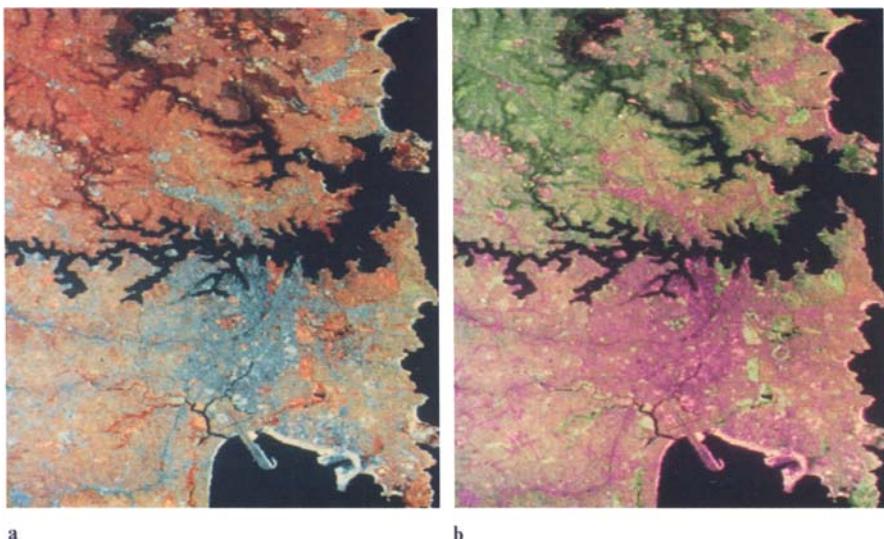
It is of interest to note that the correlation matrix for the image of Fig. 3.1 is

	Band 4	Band 5	Band 6	Band 7
Band 4	1.00			
Band 5	0.85	1.00		
Band 6	0.31	0.39	1.00	
Band 7	-0.09	-0.07	0.86	1.00

wherein the redundancy present in band 6 can be seen (see Sect. 6.1).

In many ways the choice of colour assignments for the Landsat multispectral scanner bands is an unfortunate one since this yields, for most scenes, an image product substantially composed of reds and blues. These are the hues in which the human visual system is least sensitive to detail. Instead it would have been better to form an image in which yellows and greens predominate since then many fine details become more apparent. An illustration of this is given in Fig. 3.2.

This raises a more general point about the best use of colour image display. Usually, when the bands are significantly correlated it is difficult to obtain richly coloured images in which a full range of hues is present. In order to achieve good use of the available colour space an image transformation is often required, such as the principal components transform of Chap. 6. The discussion in Sect. 6.1.1 is relevant in this regard.



**Fig. 3.2.** Standard Landsat multispectral scanner false colour composite **a** compared with a product in which band 7 has been displayed as green, band 5 as red and band 4 as blue **b**. Finer detail is more apparent in the second product owing to the sensitivity of the human vision system to yellow-green hues. The image segment shown is Sydney, the capital of New South Wales, Australia, acquired on December 14, 1980

Colour composite products for other remote sensing image data sets are created, similarly, by associating bands with display colour primaries usually in a wavelength monotonic fashion.

### 3.3

## Computer Processing for Photointerpretation

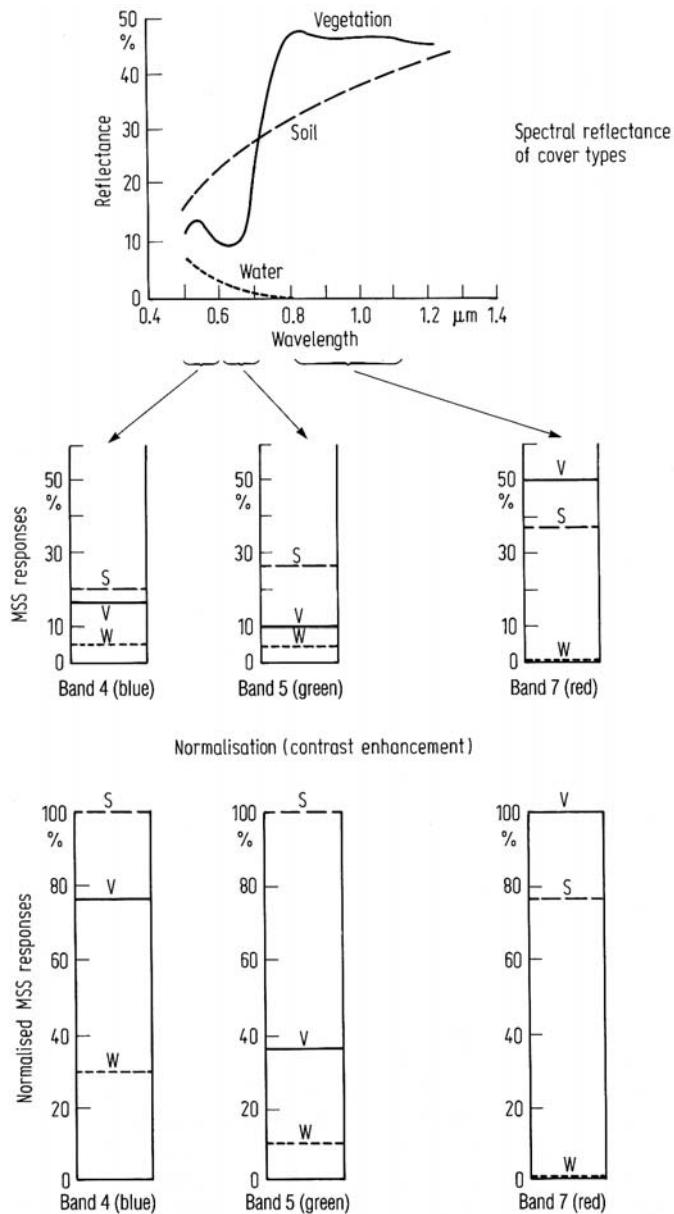
When image data is available in digital form it can be processed before an actual image is produced in order to ensure that the clues used for photointerpretation are enhanced. Little can be done about temporal clues, but judicious processing makes spectral and spatial data more meaningful. This processing is of two types. One deals with the radiometric (or brightness value) character of the image and is termed radiometric enhancement. The other has to do with the image's perceived spatial or geometric character and is referred to as geometric enhancement. The latter normally involves such operations as smoothing noise present in the data, enhancing and highlighting edges, and detecting and enhancing lines. Radiometric enhancement is concerned with altering the contrast range occupied by the pixels in an image. From the point of view of computation, radiometric enhancement procedures involve determining a new brightness value for a pixel (by some specified algorithm) from its existing brightness value. They are often referred to therefore as point operations and can be effectively implemented using look up tables. These are two-column tables that associate a set of new brightness values with the set of old brightnesses. Specific radiometric enhancement techniques are treated in Chap. 4.

Geometric enhancement procedures involve establishing a new brightness value for a pixel by using the existing brightnesses of pixels over a specified neighbourhood. The range of geometric enhancement techniques commonly used in the treatment of remote sensing image data is given in Chap. 5. Both radiometric and geometric enhancement can be of value in highlighting spatial information. It is generally only radiometric or contrast enhancement, however, that amplifies an image's spectral character. A word of caution however is in order here. When contrast enhancement is utilised, each feature in the data is generally treated independently. This can lead to a loss of feature-to-feature relativity and thus, in the case of colour composites, can lead to loss of colour relativity. The reason for this is depicted in Fig. 3.3, and the effect is illustrated in Fig. 3.4.

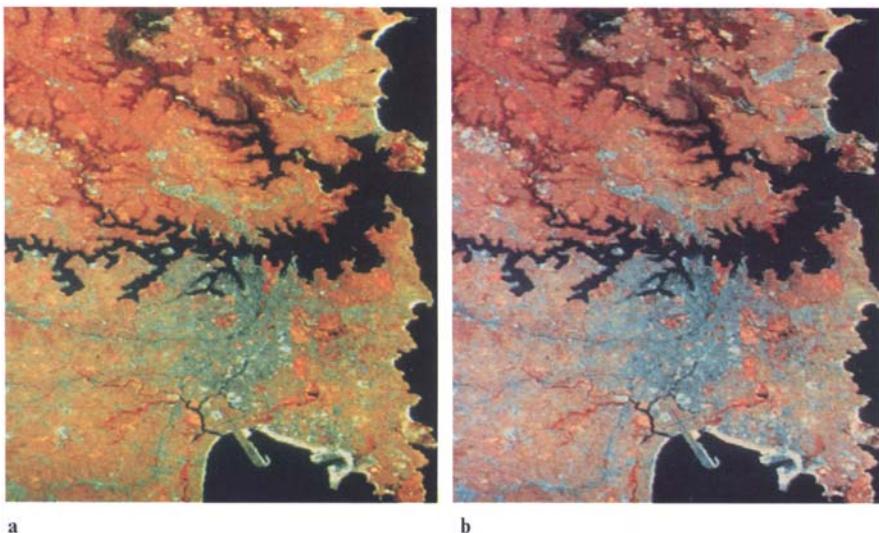
### 3.4

## An Introduction to Quantitative Analysis – Classification

Identification of features in remote sensing imagery by photointerpretation is effective for global assessment of geometric characteristics and general appraisal of ground cover types. It is, however, impracticable to apply at the pixel level unless only a handful of pixels is of interest. As a result it is of little value for determining



**Fig. 3.3.** Indication of how contrast enhancement can distort the feature-to-feature or band-to-band relativity (and thus colour relativity) in an image. Without contrast enhancement both soil and vegetation cover types would have a reddish appearance, whereas after enhancement the soil takes on its characteristic bluish tones. The bands indicated correspond to the Landsat MSS; the same effect will occur with similar band combinations from other sensors (eg. SPOT bands 1, 2 and 3)



**Fig. 3.4.** Image in which each band has identical contrast enhancement before colour composition **a** compared to that in which each band has been enhanced independently to cover its full brightness range **b**. This causes a loss of band to band relativity and thus gives a different range of hues

accurate estimates of the area in an image corresponding to a particular ground cover type, such as the hectarage of a crop. Moreover as noted in Sect. 3.2, since photointerpretation is based upon the ability of the human analyst-interpreter to assimilate the available data, only three or so of the complete set of spectral components of an image can be used readily. Yet there are seven bands in Landsat thematic mapper data and many for imaging spectrometer data. It is not that all of these would necessarily need to be used in the identification of a pixel; rather, should they all require consideration or evaluation, then the photointerpretive approach is clearly limited. Furthermore the human analyst is unable to discriminate to the limit of the radiometric resolution generally available. By comparison if a computer can be used for analysis, it could conceivably do so at the pixel level and could examine and identify as many pixels as required. In addition, it should be possible for computer analysis of remote sensing image data to take full account of the multidimensional aspect of the data including its full radiometric resolution.

Computer interpretation of remote sensing image data is referred to as quantitative analysis because of its ability to identify pixels based upon their numerical properties and owing to its ability for counting pixels for area estimates. It is also generally called classification, which is a method by which labels may be attached to pixels in view of their spectral character. This labelling is implemented by a computer by having trained it beforehand to recognise pixels with spectral similarities.

Clearly the image data for quantitative analysis must be available in digital form. This is an advantage with image data types, such as that from Landsat, SPOT, IRS,

etc, as against more traditional aerial photographs. The latter require digitisation before quantitative analysis can be performed.

Detailed procedures and algorithms for quantitative analysis are the subject of Chaps. 8, 9 and 10; Chap. 11 is used to show how these are developed into classification methodologies for effective quantitative analysis. The remainder of this chapter however is used to provide an outline of the essential concepts in classification. As a start it is necessary to devise a model with which to represent remote sensing multispectral image data in a form amenable to the development of analytical procedures.

The material in the following section assumes that we are basing our quantitative analysis on data described by a small number of bands – say no more than 10. For larger numbers, as in the case of imaging spectrometers, it may be necessary to perform feature selection first, using the procedures of Chap. 9. Otherwise, library searching or analytical methods based on spectroscopic understanding could be used, as discussed in Chap. 13.

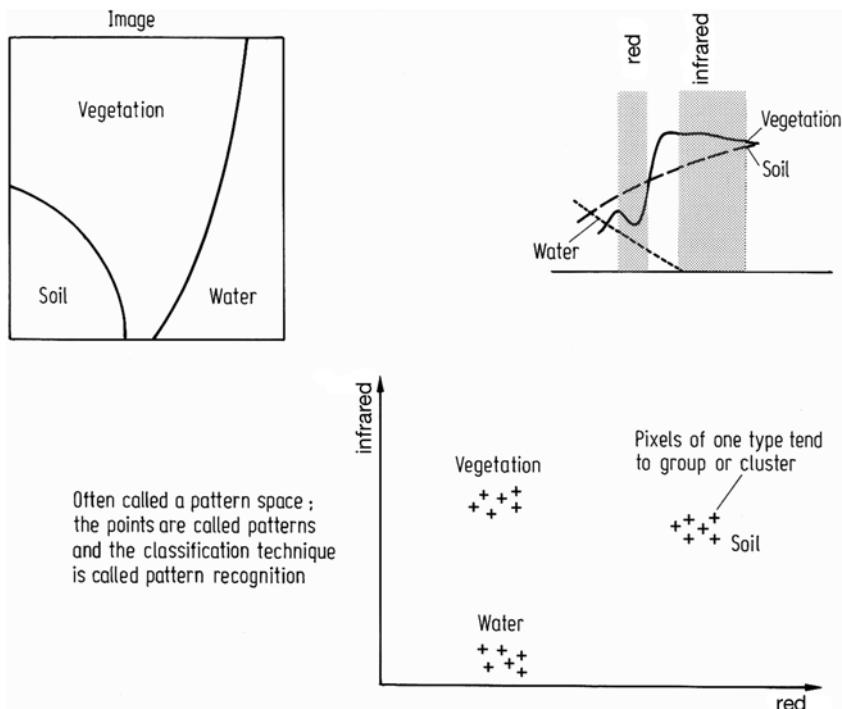
### 3.5

## Multispectral Space and Spectral Classes

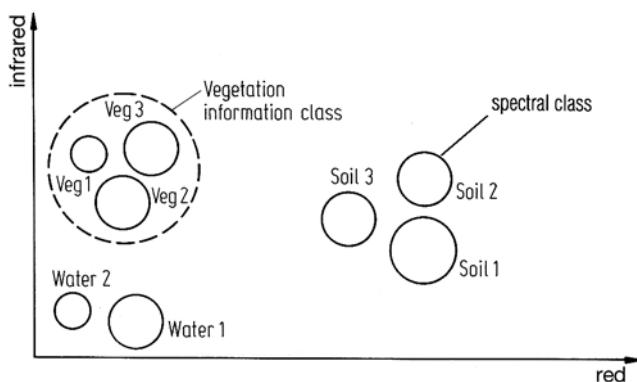
The most effective means by which multispectral data can be represented in order to formulate algorithms for quantitative analysis is to plot them in a pattern space, or multispectral vector space, with as many dimensions as there are spectral components. In this space, each pixel of an image plots as a point with co-ordinates given by the brightness value of the pixel in each component. This is illustrated in Fig. 3.5 for a simple two dimensional infrared versus visible red space. Provided the spectral bands have been designed to provide good discrimination it is expected that pixels would form groups in multispectral space corresponding to various ground cover types, the sizes and shapes of the groups being dependent upon varieties of cover type, systematic noise and topographic effects. The groups or clusters of pixel points are referred to as *information classes* since they are the actual classes of data which a computer will need to be able to recognise.

In practice the information class groupings may not be single clusters as depicted in Fig. 3.5. Instead it is not unusual to find several clusters for the same region of soil, for the same apparent type of vegetation and so on for other cover types in a scene. These are not only as a result of specific differences in types of cover but also result from differences in moisture content, soil types underlying vegetation and topographic influences. Consequently, a multispectral space is more likely to appear as shown in Fig. 3.6 in which each information class is seen to be composed of several *spectral classes*.

In many cases the information classes of interest do not form distinct clusters or groups of clusters but rather are part of a continuum of data in the multispectral space. This happens for example when, in a land systems exercise, there is a gradation of canopy closure with position so that satellite or aircraft sensors might see a gradual variation in the mixture of canopy and understory. The information classes here might



**Fig. 3.5.** Illustration of a two dimensional multispectral space showing its relation to the spectral reflectance characteristics of ground cover types



**Fig. 3.6.** Representation of information classes by sets of spectral classes

correspond to nominated percentage mixtures rather than to sets of well defined subclasses as depicted in Fig. 3.6. It is necessary in situations such as this to determine appropriate sets of spectral classes that represent the information classes effectively. This is demonstrated in the exercises chosen in Chap. 11.

In quantitative analysis it is the spectral classes that a computer will be asked to work with since they are the “natural” groupings or clusters in the data. After quantitative analysis is complete the analyst simply associates all the relevant spectral classes with the one appropriate information class. In the context of the most commonly adopted approach to classification, based on statistical models, spectral classes will be seen to be unimodal probability distributions and information classes as possible multimodal distributions. The latter need to be resolved into sets of single modes for convenience and accuracy in analysis.

## 3.6

### Quantitative Analysis by Pattern Recognition

#### 3.6.1

##### Pixel Vectors and Labelling

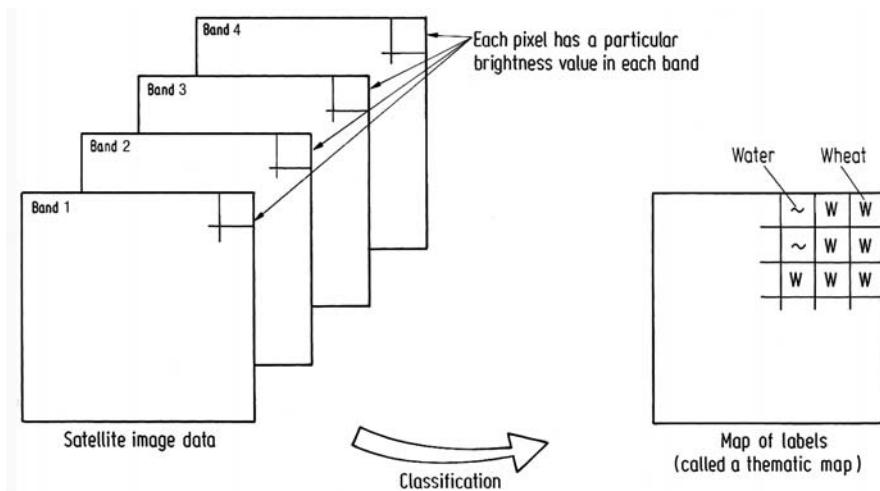
Recognition that image data exists in sets of spectral classes, and identification of those classes as corresponding to specific ground cover types, is carried out using the techniques of mathematical pattern recognition or pattern classification and their more recent machine learning variants. The patterns are the pixel themselves, or strictly the mathematical *pixel vectors* that contain the sets of brightness values for the pixels arranged in column form:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

where  $x_1$  to  $x_N$  are the brightnesses of the pixel  $\mathbf{x}$  in bands 1 to  $N$  respectively. It is simply a mathematical convention that these are arranged in a column and enclosed in an extended square bracket. A summary of essential results from the algebra used for describing and manipulating these vectors is given in Appendix D.

Classification involves labelling the pixels as belonging to particular spectral (and thus information) classes using the spectral data available. This is depicted as a mapping in Fig. 3.7. In the terminology of statistics this is more properly referred to as allocation rather than classification. However throughout this book, classification, categorization, allocation and labelling are generally used synonymously.

There are two broad classes of classification procedure and each finds application in the analysis of remote sensing image data. One is referred to as supervised classification and the other unsupervised classification. These can be used as alternative approaches but are often combined into hybrid methodologies as demonstrated in Chap. 11.



**Fig. 3.7.** The role of classification in labelling pixels in remote sensing image data

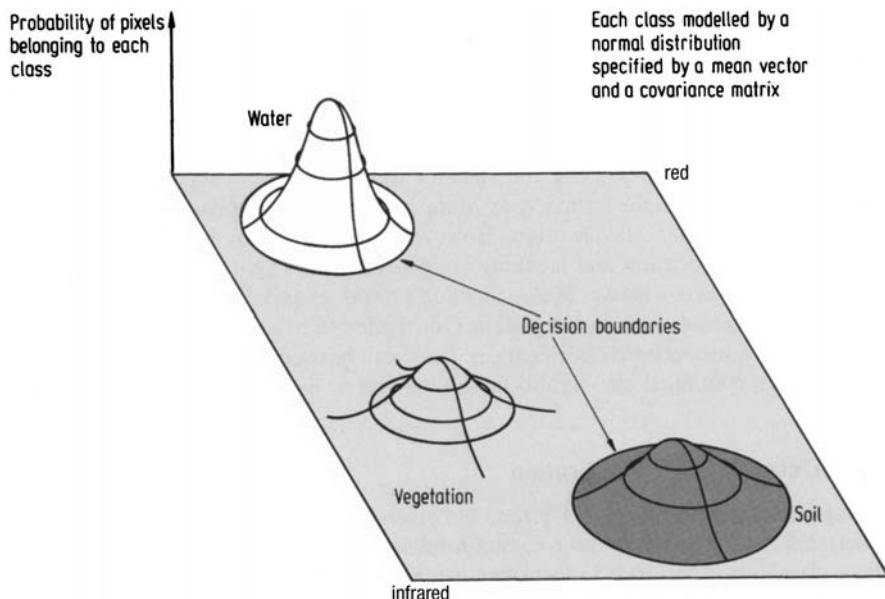
### 3.6.2 Unsupervised Classification

Unsupervised classification is a means by which pixels in an image are assigned to spectral classes without the user having foreknowledge of the existence or names of those classes. It is performed most often using clustering methods. These procedures can be used to determine the number and location of the spectral classes into which the data falls and to determine the spectral class of each pixel. The analyst then identifies those classes afterwards by associating a sample of pixels in each class with available reference data, which could include maps and information from ground visits. Clustering procedures are generally computationally expensive yet they are central to the analysis of remote sensing imagery. While the information classes for a particular exercise are known, the analyst is usually totally unaware of the spectral classes, or sub-classes as they are sometimes called. Unsupervised classification is therefore useful for determining the spectral class composition of the data prior to detailed analysis by the methods of supervised classification.

The range of clustering algorithms frequently used for determination of spectral classes and for unsupervised classification is treated in Chap. 9.

### 3.6.3 Supervised Classification

Before proceeding, it is important to recognise that a range of supervised classification procedures is possible. In the following we concentrate on a statistical methodology that has been the mainstay of quantitative analysis since the 1970s. Other methods are based on non-statistical, geometric techniques that seek to place separating surfaces between the classes shown in Figs. 3.5 and 3.6. Chapter 8 treats both



**Fig. 3.8.** Two dimensional multispectral space with the spectral classes represented by Gaussian probability distributions

statistical and geometric supervised classification in detail, using the material in the following as an introduction to the concepts involved.

An important assumption in statistical supervised classification usually adopted in remote sensing is that each spectral class can be described by a probability distribution in multispectral space: this will be a multivariable distribution with as many variables as dimensions of the space. Such a distribution describes the chance of finding a pixel belonging to that class at any given location in multispectral space. This is not unreasonable since it would be imagined that most pixels in a distinct cluster or spectral class would lie towards the centre and would decrease in density for positions away from the class centre, thereby resembling a probability distribution. The distribution found to be of most value is the normal or Gaussian distribution. It gives rise to tractable mathematical descriptions of the supervised classification process, and is robust in the sense that classification accuracy is not overly sensitive to violations of the assumptions that the classes are normal. A two dimensional multispectral space with the spectral classes so modelled is depicted in Fig. 3.8. The *decision boundaries* shown in the figure represent those points in multispectral space where a pixel has equal chance of belonging to two classes. The boundaries therefore partition the space into regions associated with each class; this is developed further in Sect. 8.2.4.

A multidimensional normal distribution is described as a function of a vector location in multispectral space by:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{N/2} |\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mathbf{m})^t \Sigma^{-1} (\mathbf{x} - \mathbf{m})\right\}$$

where  $\mathbf{x}$  is a vector location in the  $N$  dimensional pixel space:  $\mathbf{m}$  is the mean position of the spectral class – i.e. the position  $\mathbf{x}$  at which a pixel from the class is most likely to be found, and  $\Sigma$  is the covariance matrix of the distribution, which describes its spread directionally in the pixel space. Equation (6.2) shows how this matrix is defined; Appendix E summarises some of the important properties of this distribution.

The multidimensional normal distribution is specified completely by its mean vector and its covariance matrix. Consequently, if the mean vectors and covariance matrices are known for each spectral class then it is possible to compute the set of probabilities that describe the relative likelihoods of a pattern at a particular location belonging to each of those classes. It can then be considered as belonging to the class which indicates the highest probability. Therefore if  $\mathbf{m}$  and  $\Sigma$  are known for every spectral class in an image, every pixel in the image can be examined and labelled corresponding to the most likely class on the basis of the probabilities computed for the particular location for a pixel. Before that classification can be performed however  $\mathbf{m}$  and  $\Sigma$  are estimated for each class from a representative set of pixels, commonly called *a training set*. These are pixels which the analyst knows as coming from a particular (spectral) class. Estimation of  $\mathbf{m}$  and  $\Sigma$  from training sets is referred to as supervised learning. Supervised classification consists therefore of three broad steps. First a set of training pixels is selected for each spectral class. This may be done using information from ground surveys, aerial photography, topographic maps or any other source of reference data. The second step is to determine  $\mathbf{m}$  and  $\Sigma$  for each class from the training data. This completes the learning phase. The third step is the classification phase, in which the relative likelihoods for each pixel in the image are computed and the pixel labelled according to the highest likelihood.

The view of supervised classification adopted here has been based upon an assumption that the classes can be modelled by probability distributions and, as a consequence, are described by the parameters of those distributions. As a result it is also referred to as a parametric supervised method. Other supervised techniques also exist, in which neither distribution models nor parameters are relevant. These are referred to as non-parametric methods. More recently, neural networks and support vector machine non-parametric classification methods have been shown to offer promise in remote sensing applications, as demonstrated in Sects. 8.9.1 and 8.9.2.

## References for Chapter 3

A good summary, with extensive references, of the spectral reflectance characteristics of common earth surface cover types has been given by Hoffer (1978). Material of this type is important in photointerpretation. Landgrebe (1981) and Hoffer (1979) have provided good general discussions on computer classification of remote sensing image data.

More recent quantitative treatments will be found in Schowengerdt (1997), Landgrebe (2003) and Mather (1987). Schott (1997) has treated remote sensing data flow from a systems perspective.

- R.M. Hoffer, 1978: Biological and Physical Considerations in Applying Computer-Aided Analysis Techniques to Remote Sensing Data, in P.H. Swain & S.M. Davis, Eds.: *Remote Sensing: The Quantitative Approach*, McGraw-Hill, N.Y.
- R.M. Hoffer, 1979: Computer Aided Analysis Techniques for Mapping Earth Surface Features, Technical Report 020179, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette, Indiana.
- D.A. Landgrebe, 1981: Analysis Technology for Land Remote Sensing, Proc. IEEE, 69, 628-642.
- P.M. Mather, 1987: *Computer Processing of Remotely Sensed Images*, Wiley, Chichester. N.Y.
- D.A. Landgrebe, 2003: *Signal Theory Methods in Multispectral Remote Sensing*, N.J., Wiley.
- P.M. Mather, 1987: *Computer Processing of Remotely Sensed Images*, Wiley, Chichester.
- J.R. Schott, 1997: *Remote Sensing: The Image Chain Approach*, Oxford UP, N.Y.
- R.A. Schowengerdt, 1997: *Remote Sensing Models and Methods for Image Processing*, 2e, Academic, MA.

## Problems

**3.1** For each of the following applications would photointerpretation or quantitative analysis be the most appropriate analytical technique? Where necessary, assume spectral discrimination is possible.

- (i) Lithological mapping in geology
- (ii) Structural mapping in geology
- (iii) Assessment of forest condition
- (iv) Mapping movements of floods
- (v) Crop area determination
- (vi) Crop health assessment
- (vii) Bathymetric charting
- (viii) Soil mapping
- (ix) Mapping drainage patterns
- (x) Land system mapping

**3.2** Can contrast enhancing image data beforehand improve its discrimination for machine analysis?

**3.3** Prepare a table comparing the attributes of supervised and unsupervised classification. You may care to consider the issues of training data, cost (see Chap. 11), analyst interaction and spectral class determination.

**3.4** A problem with using probability models to describe classes in multispectral space is that atypical pixels can be erroneously classified. For example, a pixel with high red and infrared brightness in Fig. 3.8 would be classified as vegetation even though it is more reasonably soil. This is a result of the positions of the decision boundaries shown. Suggest a means by which this situation can be avoided. (This is taken up in Sect. 8.2.5).

**3.5** The collection of the four brightness values for a pixel in a Landsat multispectral scanner image is often called a vector. Each of the four components in such a vector can take either

128 or 64 different values, depending upon the band. How many distinct pixel vectors are possible? How many are there for Landsat thematic mapper image data?

It is estimated that the human visual system can discriminate about 20,000 colours (J.O. Whittaker, Introduction to Psychology, Saunders, Philadelphia, 1965).

Comment on the radiometric handling capability of machine analysis, compared to colour discrimination by a human analyst/interpreter.

**3.6** Information classes are resolved into so-called spectral classes prior to classification. These are pixel groups amenable to modelling by single multivariate Gaussian or normal distribution functions. Why are more complex distributions not employed to obviate the need to establish spectral classes? (Hint: How much is known about *multi-variate* distributions other than Gaussian?)

# 4

## Radiometric Enhancement Techniques

### 4.1

#### Introduction

##### 4.1.1

##### Point Operations and Look Up Tables

Image analysis by photointerpretation is often facilitated when the radiometric nature of the image is enhanced to improve its visual impact. Specific differences in vegetation and soil types, for example, may be brought out by increasing the contrast of an image. In a similar manner subtle differences in brightness value can be highlighted either by contrast modification or by assigning quite different colours to those levels. The latter method is known as colour density slicing.

It is the purpose of this chapter to present a variety of radiometric modification procedures often used with remote sensing image data. The range of techniques treated is characterised by the common feature that a new brightness value for a pixel is generated only from its existing value. Neighbouring pixels have no influence, as they do in the geometric enhancement procedures that are the subject of Chap. 5. Consequently, radiometric enhancement techniques are sometimes referred to as point or pixel-specific operations.

All of the techniques to be covered in this chapter can be represented either as a graph or as a table that expresses the relationship between the old and new brightness values. In tabular form this is referred to as a look up table (LUT).

##### 4.1.2

##### Scalar and Vector Images

Two particular image types require consideration when treating image enhancement. The first could be referred to as a *scalar* image, in which each pixel has only a single brightness value associated with it. Such is the case for a simple black and white image. The second type is a *vector* image, in which each pixel is represented by

a vector of brightness values, which might be the blue, green and red components of the pixel in a colour scene or, for a remote sensing multispectral image, may be the various spectral response components for the pixel. Most image enhancement techniques relate to scalar images and also to the scalar components of vector imagery. Such is the case with all techniques given in this chapter. Enhancement methods that relate particularly to vector imagery tend to be transformation oriented. Those are treated in Chap. 6.

## 4.2

### The Image Histogram

Consider a spatially quantised scalar image such as that corresponding to one of the Landsat thematic mapper bands; in this case the brightness values are also quantised. If each pixel in the image is examined and its brightness value noted, a graph of number of pixels with a given brightness versus brightness value can be constructed. This is referred to as the histogram of the image. The tonal or radiometric quality of an image can be assessed from its histogram as illustrated in Fig. 4.1. An image which makes good use of the available range of brightness values has a histogram with occupied bins (or bars) over its full range, but without significantly large bars at black or white.

An image has a unique histogram but the reverse is not true in general since a histogram contains only radiometric and no spatial information. A point of some importance is that the histogram can be viewed as a discrete probability distribution since the relative height of a particular bar, normalised by the total number of pixels in the image segment, indicates the chance of finding a pixel with that particular brightness value somewhere in the image.

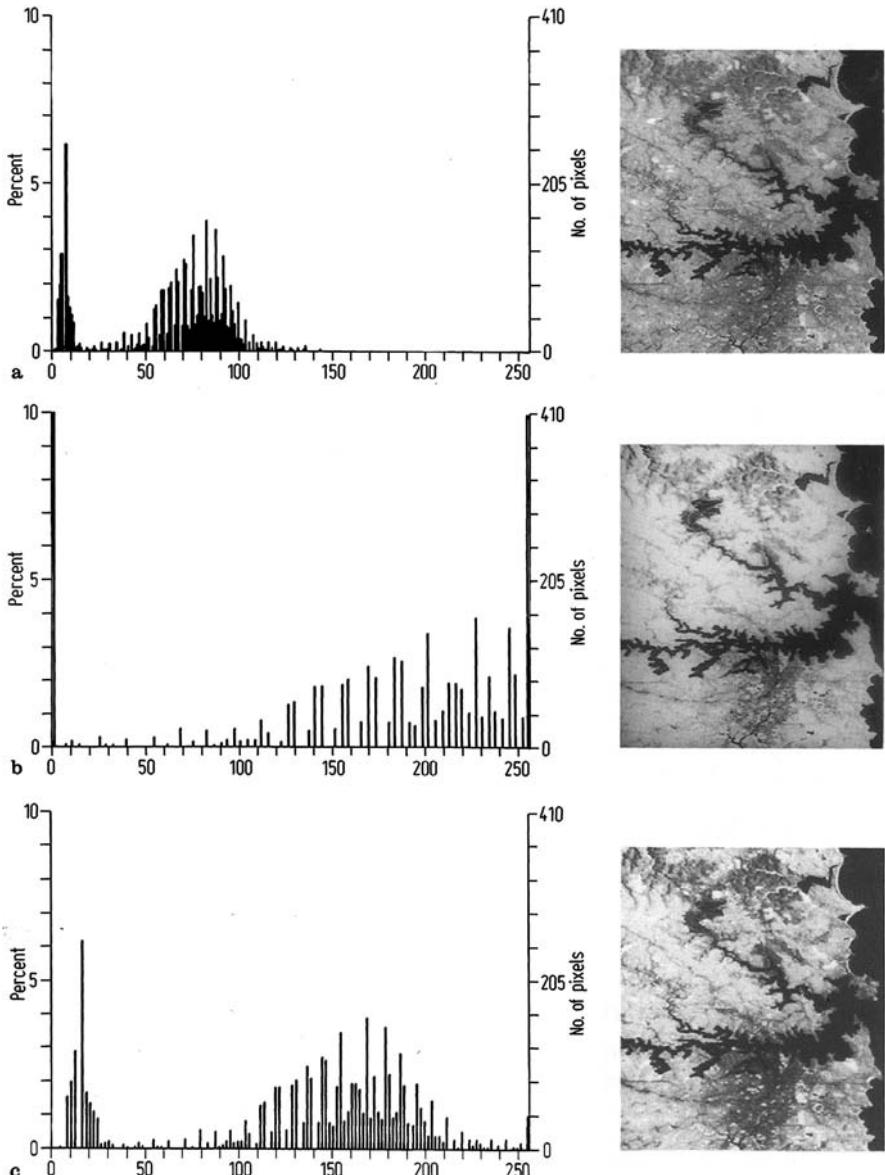
## 4.3

### Contrast Modification in Image Data

#### 4.3.1

##### Histogram Modification Rule

Suppose one has available a digital image with poor contrast, such as that in Fig. 4.1a, and it is desired to improve its contrast to obtain an image with a histogram that has a good spread of bars over the available brightness range, resembling that in Fig. 4.1c. In other words, a so-called contrast stretching of the image data is required. Often the degree of stretching desired is apparent. For example the original histogram may occupy brightness values between 40 and 75 and we might wish to expand this range to the maximum possible, say 0 to 255. Even though the modification is somewhat obvious it is necessary to express it in mathematical terms in order to relegate it to a computer. Contrast modification is a mapping of brightness values, in that the



**Fig. 4.1.** Examples of image histograms. The image in **a** shows poor contrast since its histogram utilizes a restricted range of brightness value. The image in **b** is very contrasty with saturation in the black and white regions resulting in some loss of discrimination of bright and dull features. The image in **c** makes optimum use of the available brightness levels and shows good contrast. Its histogram shows a good spread of bars but without the large bars at black and white indicative of the saturation in image **b**

brightness value of a particular histogram bar is respecified more favourably. The bars themselves though are not altered in size, although in some cases some bars may be mapped to the same new brightness value and will be superimposed. In general, however, the new histogram will have the same number of bars as the old. They will simply be at different locations.

The mapping of brightness values associated with contrast modification can be described as

$$y = f(x) \quad (4.1)$$

where  $x$  is the old brightness value of a particular bar in the histogram and  $y$  is the corresponding new brightness value.

In principle, what we want to do in contrast modification is find the form of  $f(x)$  that will implement the desired changes in pixel brightness and thus in the perceived contrast of the image. Sometimes that is quite simple; on other occasions  $f(x)$  might be quite a complicated function. In the following sections we look at simple contrast changes first.

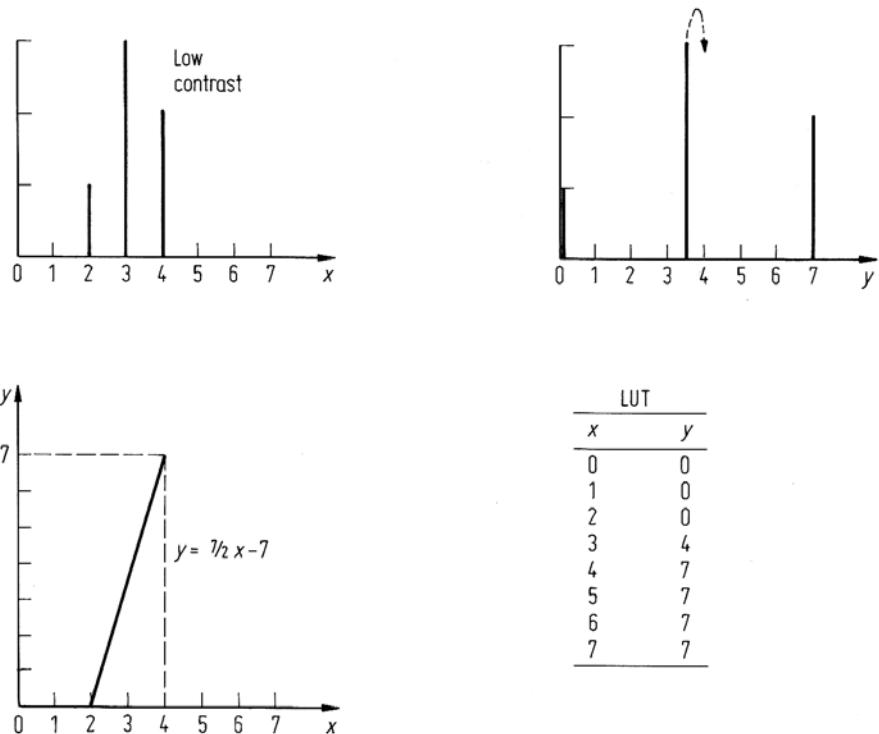
### 4.3.2 Linear Contrast Modification

The most common contrast modification operation is that in which the new ( $y$ ) and old ( $x$ ) brightness values of the pixels in an image are related in a linear fashion, i.e. so that (4.1) can be expressed

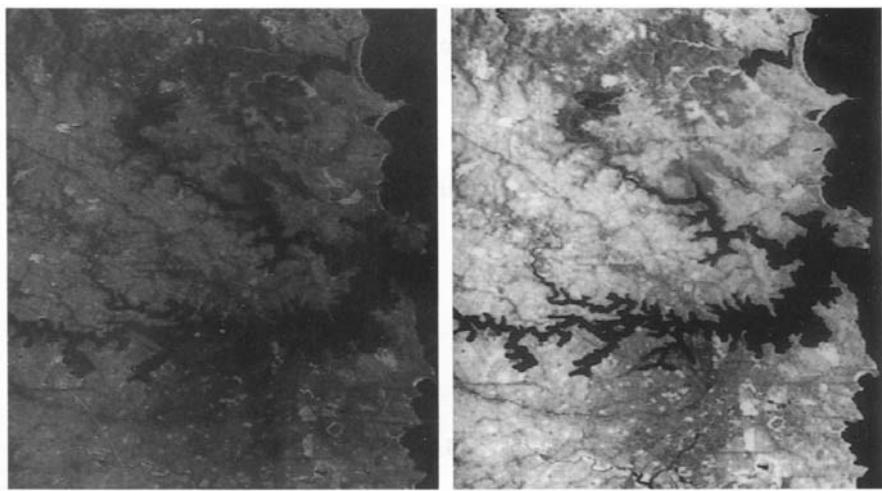
$$y = f(x) = ax + b.$$

A simple numerical example of linear contrast modification is shown in Fig. 4.2, whereas a poorly contrasting image that has been radiometrically enhanced by linear contrast stretching is shown in Fig. 4.3.

The look-up table for the particular linear stretch in Fig. 4.2 has been included in the figure. In practice this would be used by a computer routine to produce the new image. This is done by reading the brightness values of the original version, pixel by pixel, substituting these into the left hand side of the table and then reading the new brightness value for a pixel from the corresponding entry on the right hand side of the table. It is important to note in digital image handling that the new brightness values, just as the old, must be discrete, and cover usually the same range of brightnesses. Generally this will require some rounding to integer form of the new brightness values calculated from the mapping function  $y = f(x)$ . A further point to note in the example of Fig. 4.2 is that the look-up table is undefined outside the range 2 to 4 of inputs. To do so would generate output brightness values that are outside the range valid for this example. In practice, linear contrast stretching is generally implemented as the saturating linear contrast enhancement technique in Sect. 4.3.3 following.



**Fig. 4.2.** Simple numerical example of linear contrast modification. The available range of discrete brightness values is 0 to 7. Note that a non-integral output brightness value might be indicated. In practice this is rounded to the nearest integer



**Fig. 4.3.** Linear contrast modification of the image in **a** to produce the visually better product in **b**

### 4.3.3

#### Saturating Linear Contrast Enhancement

Frequently a better image product is given when linear contrast enhancement is used to give some degree of saturation at the black and white ends of the histogram. Such is the case, for example, if the darker regions in an image correspond to the same ground cover type within which small radiometric variations are of no interest. Similarly, a particular region of interest in an image may occupy a restricted brightness value range; saturating linear contrast enhancement is then employed to expand that range to the maximum possible dynamic range of the display device with all other regions being mapped to either black or white. The brightness value mapping function  $y = f(x)$  for saturating linear contrast enhancement is shown in Fig. 4.4, in which  $B_{max}$  and  $B_{min}$  are the user-determined maximum and minimum brightness values that are to be expanded to the lowest and highest brightness levels supported by the display device.

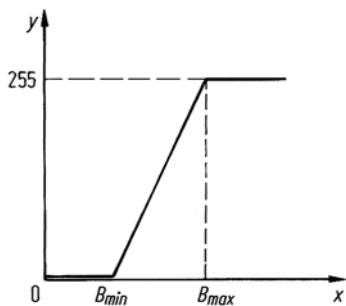


Fig. 4.4. Saturating linear contrast mapping

### 4.3.4

#### Automatic Contrast Enhancement

Most remote sensing image data is too low in brightness and poor in contrast to give an acceptable image product if displayed directly in raw form. This is a result of the need to have the dynamic range of satellite and aircraft sensors so adjusted that a variety of cover types over many images can be detected without leading to saturation of the detectors or without useful signals being lost in noise. As a consequence a single typical image will contain a restricted set of brightnesses.

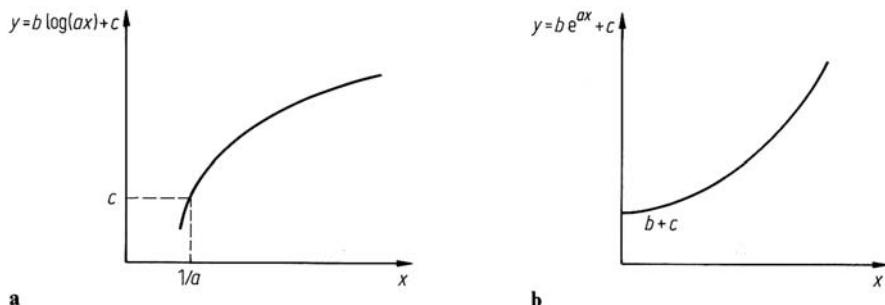
Image display systems frequently implement an automatic contrast stretch on the raw data in order to give a product with good contrast.

Typically the automatic enhancement procedure is a saturating linear stretch. The cut-off and saturation limits  $B_{min}$  and  $B_{max}$  are chosen by determining the mean brightness of the raw data and its standard deviation and then making  $B_{min}$  equal to the mean less three standard deviations and  $B_{max}$  equal to the mean plus three standard deviations.

### 4.3.5

#### Logarithmic and Exponential Contrast Enhancement

Logarithmic and exponential mappings of brightness values between original and modified images are useful for enhancing dark and light features respectively. The mapping functions are depicted in Fig. 4.5, along with their mathematical expressions. It is particularly important with these that the output values be scaled to lie within the range of the device used to display the product (or the range appropriate to files used for storage in a computer memory) and that the output values be rounded to allowed, discrete values.

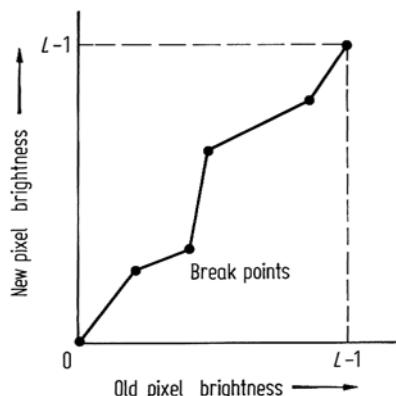


**Fig. 4.5.** Logarithmic **a** and exponential **b** brightness mapping functions. The parameters  $a$ ,  $b$  and  $c$  are usually included to adjust the overall brightness and contrast of the output product

### 4.3.6

#### Piecewise Linear Contrast Modification

A particularly useful and flexible contrast modification procedure is the piecewise linear mapping function shown in Fig. 4.6. This is characterised by a set of user specified break points as shown. Generally the user can also specify the number of



**Fig. 4.6.** Piecewise linear contrast modification function, characterised by the break points shown. These are user specified (as new, old pairs). It is clearly important that the function commence at  $0,0$  and finish at  $L - 1, L - 1$  as shown, where  $L$  is the total number of brightness levels

break points. This method has particular value in implementing some of the contrast matching procedures in Sects. 4.4 and 4.5 following.

It should be noted that this is a more general version of the saturating linear contrast stretch of Sect. 4.3.3.

## 4.4

### Histogram Equalization

#### 4.4.1

##### Use of the Cumulative Histogram

The foregoing sections have addressed the task of simple expansion (or contraction) of the histogram of an image. In many situations however it is desirable to modify the contrast of an image so that its histogram matches a preconceived shape, other than a simple closed form mathematical modification of the original version. A particular and important modified shape is the uniform histogram in which, in principle, each bar has the same height. Such a histogram has associated with it an image that utilises the available brightness levels equally and thus should give a display in which there is good representation of detail at all brightness values. In practice a perfectly uniform histogram cannot be achieved for digital image data; the procedure following however produces a histogram that is quasi-uniform on the average. The method of producing a uniform histogram is known generally as histogram equalization.

It is useful, in developing the actual methods to be used for histogram equalisation, if we regard the histograms as continuous curves as depicted in Fig. 4.7, adapted from Castleman (1996). In this  $h_i(x)$  represents the original image histogram (the “input” to the modification process) and  $h_o(y)$  represents the histogram of the image after it has had its contrast modified (the “output” from the modification process).

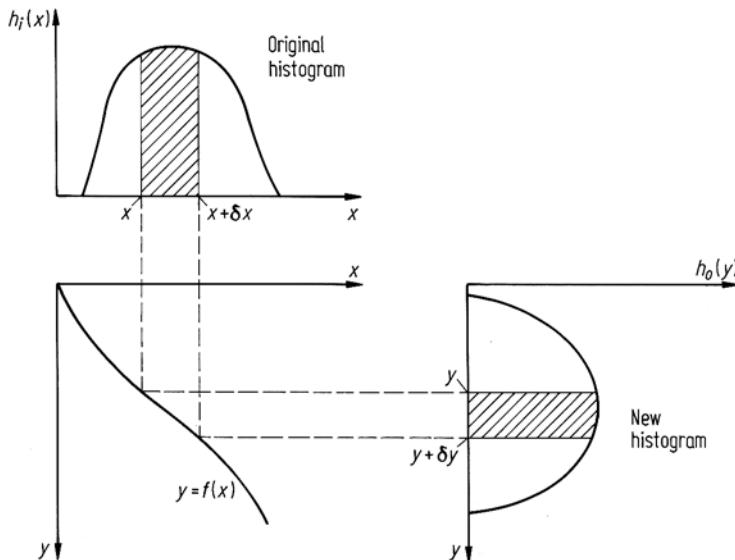
In Fig. 4.7 the number of pixels represented by the range  $y$  to  $y + \delta y$  in the modified histogram must, by definition in the diagram, be equal to the number of pixels represented in the range  $x$  to  $x + \delta x$  in the original histogram. Given that  $h_i(x)$  and  $h_o(y)$  are strictly density functions, this implies

$$h_i(x)\delta x = h_o(y)\delta y$$

so that in the limit as  $\delta x$ ,  $\delta y \rightarrow 0$ , using simple calculus

$$h_o(y) = h_i(x) \frac{dx}{dy} \quad (4.2)$$

We can use this last expression in two ways. First, if we know the original (input) histogram – which is usually always the case – and the function  $y = f(x)$ , we can determine the resulting (output) histogram. Alternatively, if we know the original histogram, and the shape of the output histogram we want – e.g. “flat” in the case of contrast equalisation – then we can use (4.2) to help us find the  $y = f(x)$  that will generate that result. Our interest here is in the second approach.



**Fig. 4.7.** Diagrammatic representation of contrast modification by the brightness value mapping function  $y = f(x)$

Note that if  $y = f(x)$ , and thus  $x = f^{-1}(y)$ , (4.2) can be expressed

$$h_o(y) = h_i(f^{-1}(y)) \frac{df^{-1}(y)}{dy}$$

which is a mathematical expression for the modified histogram<sup>1,2</sup>.

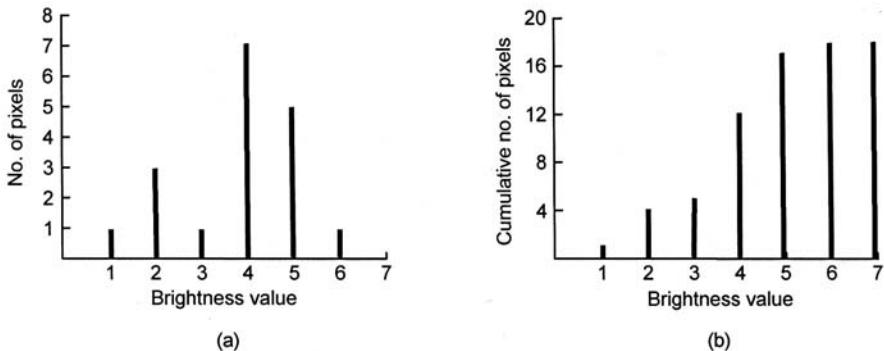
To develop the brightness value modification procedure for contrast equalisation it is convenient to re-express (4.2) as

$$\frac{dy}{dx} = \frac{h_i(x)}{h_o(y)}$$

For a uniform histogram  $h_o(y)$  and thus  $1/h_o(y)$  should be constant – i.e. independent of  $y$ . This is a mathematical idealisation for real data, and rarely will we achieve a totally flat modified histogram, as the examples in the following will show. However,

<sup>1</sup> This requires the inverse  $x = f^{-1}(y)$  to exist. For the contrast modification procedures used in remote sensing that is generally the case. Should an inverse not exist – for example if  $y = f(x)$  is not monotonic – Castleman (1996) recommends treating the original brightness value range  $x$  as a set of contiguous sub-ranges within each of which  $y = f(x)$  is monotonic.

<sup>2</sup> If we apply this expression to the brightness value modification function for linear contrast enhancement we have  $y = ax + b$ , giving  $x = \frac{y-b}{a}$  so that  $h_o(y) = \frac{1}{a} h\left(\frac{y-b}{a}\right)$ . Relative to the original histogram, the modified version is shifted because of the effect of  $b$ , is spread or compressed depending on whether  $a$  is greater or less than 1 and is modified in amplitude. The last effect only relates to the continuous function and cannot happen with discrete brightness value data.



**Fig. 4.8.** a simple histogram and b the corresponding cumulative histogram

making this assumption mathematically will generate for us the process we need to adopt to equalise image histograms. With this we can write the last expression as

$$\frac{dy}{dx} = \text{constant } h_i(x)$$

so that

$$dy = \text{constant } h_i(x)dx$$

giving by integration

$$y = \text{constant } \int h_i(x)dx .$$

How should we interpret the integral on the right hand side of this last expression? In effect it is the continuous version of a cumulative histogram which, in discrete form, is a graph of the number of pixels below a given brightness value as a function of brightness value as illustrated in Fig. 4.8. The cumulative histogram is computed by summing the bars of the ordinary histogram from left to right.

If we call the cumulative histogram  $C(x)$ , then

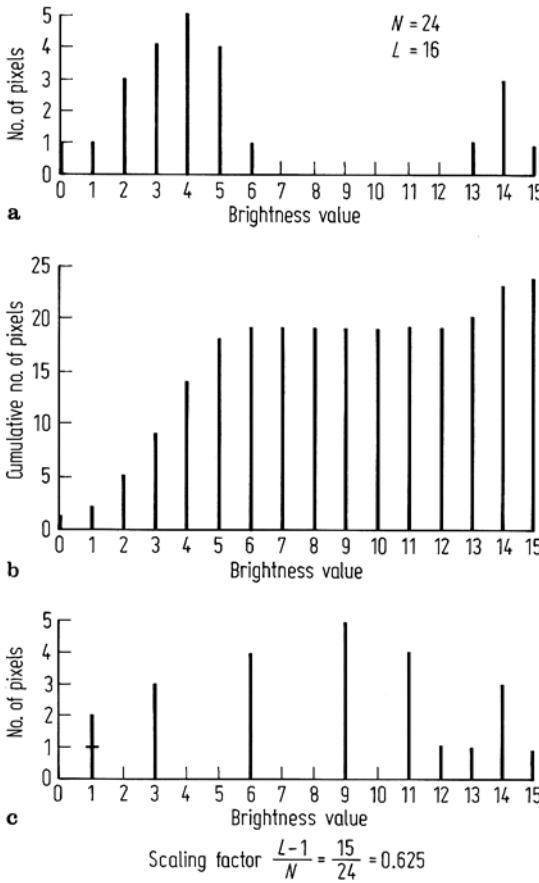
$$y = \text{constant } C(x)$$

is the brightness value modification formula for histogram (contrast) equalisation. How do we find the value of the “constant”? We note first that the range of values of  $y$  is required to be 0 to  $L - 1$  to match the  $L$  brightness values available in the image. Secondly, note that the maximum value of  $C(x)$  is  $N$ , the total number of pixels in the image, as seen in Fig. 4.8. Thus the constant needs to be  $(L - 1)/N$  in order to generate the correct range for  $y$ . In summary, the brightness value mapping function that gives contrast equalisation is

$$y = \frac{L - 1}{N} C(x) . \quad (4.3)$$

where  $C(x)$  is the discrete cumulative histogram.

Equation (4.3) is, in effect, a look-up table that can be used to move histogram bars to new brightness value locations. To illustrate the concept, consider the need to

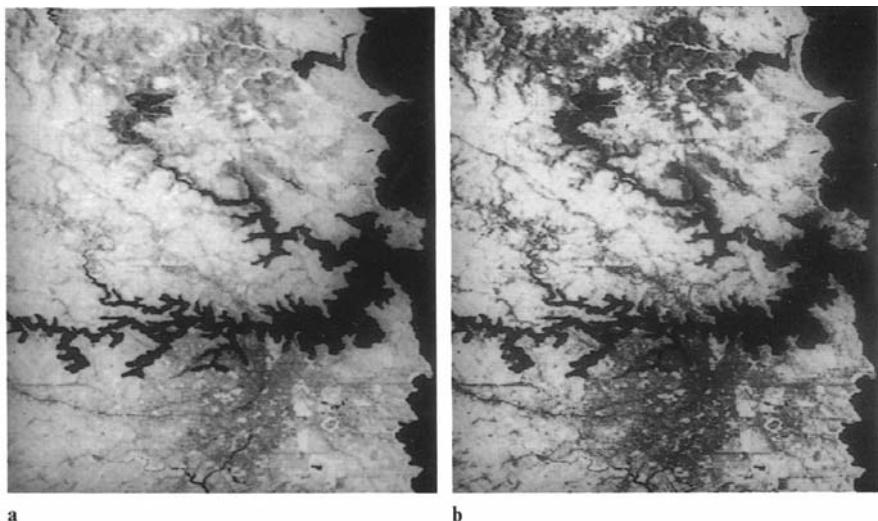


**Fig. 4.9.** Example of histogram equalisation. **a** Original histogram; **b** Cumulative histogram used to produce the look up table in Table 4.1; **c** The resulting quasi-uniform histogram

“flatten” the simple histogram shown in Fig. 4.9a. This corresponds to a hypothetical image with 24 pixels, each of which can take on one of 16 possible brightness values. The corresponding cumulative histogram is shown in Fig. 4.9b, and the scaling factor in (4.3) is  $(L - 1)/N = 15/24 = 0.625$ . Using (4.3) the new brightness value location of a histogram bar is given by finding its original location on the abscissa of the cumulative histogram ( $x$ ) and then reading its unscaled new location ( $y$ ) from the ordinate. Multiplication by the scaling factor then produces the required new value. It is likely, however, that this may not be one of the discrete brightness values available (for the output display device) in which case the associated bar is moved to the nearest available brightness value. This procedure is summarised, for the example at hand, in Table 4.1, and the new, quasi-uniform histogram is given in Fig. 4.9c. It is important to emphasise that additional brightness values cannot be created nor can pixels from a single brightness value in an original histogram be distributed over several brightness values in the modified version. All that can be done is to re-map the brightness values to give a histogram that is as uniform as possible. Sometimes

**Table 4.1.** Look up table generation for histogram equalization example

Original brightness value	Unscaled new value	Scaled new value	Nearest available brightness value
0	1	0.63	1
1	2	1.25	1
2	5	3.13	3
3	9	5.63	6
4	14	8.75	9
5	18	11.25	11
6	19	11.88	12
7	19	11.88	12
8	19	11.88	12
9	19	11.88	12
10	19	11.88	12
11	19	11.88	12
12	19	11.88	12
13	20	12.50	13
14	23	14.40	14
15	24	15.00	15

**Fig. 4.10.** Image with linear contrast stretch **a** compared with the same image enhanced with a stretch from histogram equalization **b**

this entails some bars from the original histogram being moved to the same new location and thereby being superimposed, as is observed in the example.

In practice, the look up table created in Table 4.1 would be applied to every pixel in the image by feeding into the table the original brightness value for the pixel and reading from the table the new brightness value.

Figure 4.10 shows an example of an image with a simple linear contrast modification compared to the same image but in which contrast modification by histogram

**Table 4.2.** Look up table for histogram equalization using 8 output brightnesses from 16 input brightnesses

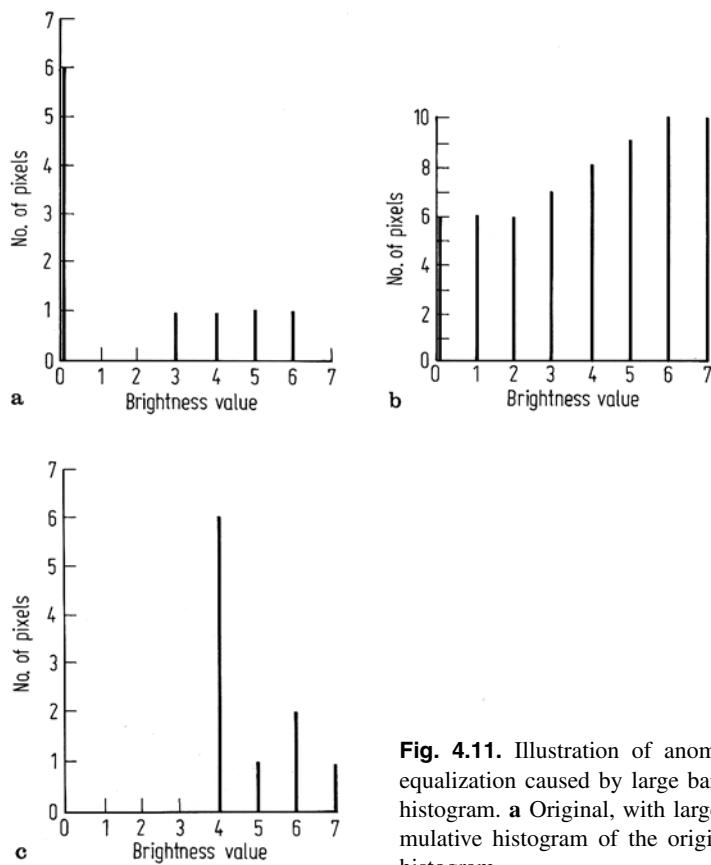
Original brightness value	Unscaled new value	Scaled new value	Nearest available brightness value
0	1	0.29	0
1	2	0.58	1
2	5	1.46	1
3	9	2.63	3
4	14	4.08	4
5	18	5.25	5
6	19	5.54	6
7	19	5.54	6
8	19	5.54	6
9	19	5.54	6
10	19	5.54	6
11	19	5.54	6
12	19	5.54	6
13	20	5.83	6
14	23	6.70	7
15	24	7.00	7

equalization has been implemented. Many of these subtle contrast changing techniques only give perceived improvement of detail on some image types and sometimes require all components of a colour composite image to be so processed before an “improvement” is noticeable.

It is not necessary to retain the same number of distinct brightness values in an equalized histogram as in the original. Sometimes it is desirable to have a smaller output set and thereby produce a histogram with (fewer) bars that are closer in height than would otherwise be the case. This is implemented by redefining  $L$  in (4.3) to be the new total number of bars. Repeating the example of Table 4.1 and Fig. 4.9 for the case of  $L = 8$  (rather than 16) gives the look up table of Table 4.2. Such a strategy would be an appropriate one to adopt when using an output device with a small number of brightness values (grey levels).

#### 4.4.2 Anomalies in Histogram Equalization

Images with extensive homogeneous regions will give rise to histograms with large bars at the corresponding brightness values. A particular example is a Landsat multispectral scanner infrared image with a large expanse of water. Because histogram equalization creates a histogram that is uniform on the average by grouping smaller bars together, the equalized version of an image such as that just described will have poor contrast and little detail – quite the opposite to what is intended. The reason for this can be seen in the simple illustration of Fig. 4.11. The cumulative histogram used as the look-up table for the enhancement is dominated by the large bar at brightness value 0. The resulting image would be mostly grey and white with little grey level discrimination.



**Fig. 4.11.** Illustration of anomalous histogram equalization caused by large bars in the original histogram. **a** Original, with large bar at 0; **b** Cumulative histogram of the original; **c** Equalized histogram

A similar situation happens when the automatic contrast enhancement procedure of Sect. 4.3.4 is applied to images with large regions of constant brightness. This can give highly contrasting images on colour display systems; an acceptable display may require some manual adjustment of contrast taking due regard of the abnormally large histogram bars.

To avoid the anomaly in histogram equalization caused by the types of image discussed it is necessary to reduce the significance of the dominating bars in the image histograms. This can be done simply by arbitrarily reducing their size when constructing the look up table, remembering to take account of this in the scale factor of (4.3). Another approach is to produce the cumulative histogram and thus look-up table on a subset of the image that does not include any, or any substantial portion, of the dominating region. Hogan (1981) has also provided an alternative procedure, based upon accumulating the histogram over “buckets” of brightness value. Once a bucket is full to a prespecified level, a new bucket is started.

## 4.5

# Histogram Matching

### 4.5.1

#### Principle of Histogram Matching

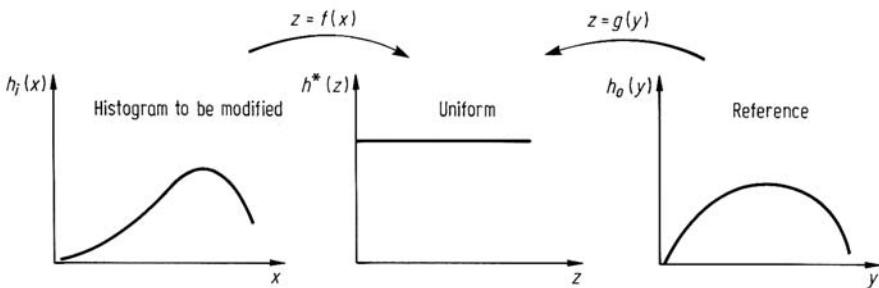
Frequently it is desirable to match the histogram of one image to that of another image and in so doing make the apparent distribution of brightness values in the two images as close as possible. This would be necessary for example when a pair of contiguous images are to be joined to form a mosaic. Matching their histograms will minimise the brightness value variations across the join. In another case, it might be desirable to match the histogram of an image to a pre-specified shape, other than the uniform distribution treated in the previous section. For example, it is often found of value in photointerpretation to have an image whose histogram is a Gaussian function of brightness, in which most pixels have mid-range brightness values with only a few in the extreme white and black regions. The histogram matching technique, to be derived now, allows both of these procedures to be implemented.

The process of histogram matching is best looked at as having two stages, as depicted in Fig. 4.12. Suppose it is desired to match the histogram of a given image,  $h_i(x)$ , to the histogram  $h_o(y)$ ;  $h_o(y)$  could be a pre-specified mathematical expression or the histogram of the second image. Then the steps in the process are to equalize the histogram  $h_i(x)$  by the methods of the previous section to obtain an intermediate histogram  $h^*(z)$ , which is then modified to the desired shape  $h_o(y)$ .

If  $z = f(x)$  is the transformation that flattens  $h_i(x)$  to produce  $h^*(z)$  and  $z = g(y)$  is the operation that would flatten the reference histogram  $h_o(y)$  then the overall mapping of brightness values required to produce  $h_o(y)$  from  $h_i(x)$  is

$$y = g^{-1}(z), \quad z = f(x) \quad \text{or} \quad y = g^{-1}\{f(x)\}. \quad (4.4)$$

If, as is often the case, the number of pixels and brightness values in  $h_i(x)$  and  $h_o(y)$  are the same, then the  $(L - 1)/N$  scaling factor in (4.3) will cancel in (4.4) and can therefore be ignored in establishing the look up table which implements the contrast matching process. Should the number of pixels be different, say  $N_1$  in the image to be modified and  $N_2$  in the reference image then a scaling factor of



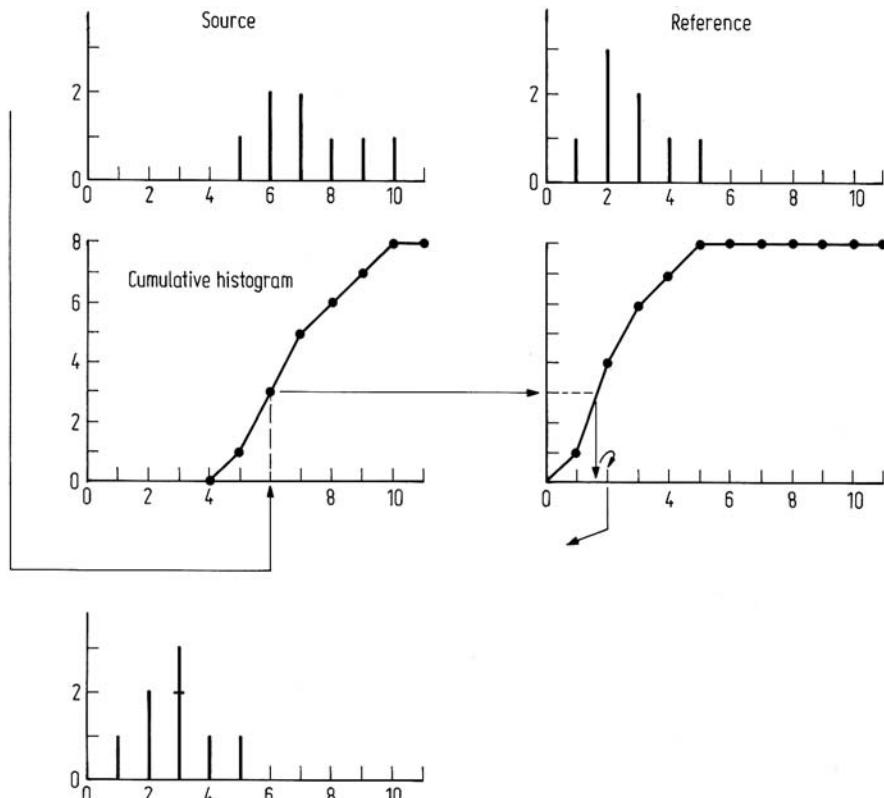
**Fig. 4.12.** The stages in histogram matching

$N_2/N_1$  will occur in (4.4). All scaling considerations can be bypassed however if the cumulative histograms are always scaled to some normalised value such as unity, or 100% (of the total number of pixels in an image).

#### 4.5.2

#### Image to Image Contrast Matching

Figure 4.13 illustrates the steps implicit in (4.4) in matching source and reference histograms. In this case the reference histogram is that of a second image. Note that the procedure is to use the cumulative histogram of the source image to obtain new brightness values in the manner of the previous section by reading ordinate values corresponding to original brightness values entered on the abscissa. The new values are then entered into the *ordinate* of the cumulative reference histogram and the final brightness values (for the bars of the source histogram) are read from the *abscissa*; i.e. the cumulative reference histogram is used in reverse as indicated by the  $g^{-1}$  operation in (4.4). The look up table for this example is shown in Table 4.3. Again,



**Fig. 4.13.** An illustration of the steps in histogram matching

**Table 4.3.** Look up table generation for contrast matching

Source histogram brightness values x	Intermediate (equalized) values z	Modified values y	Nearest available brightness values
0	0	0	0
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0
5	1	1	1
6	3	1.8	2
7	5	2.6	3
8	6	3	3
9	7	4	4
10	8	8	5
11	8	8	5

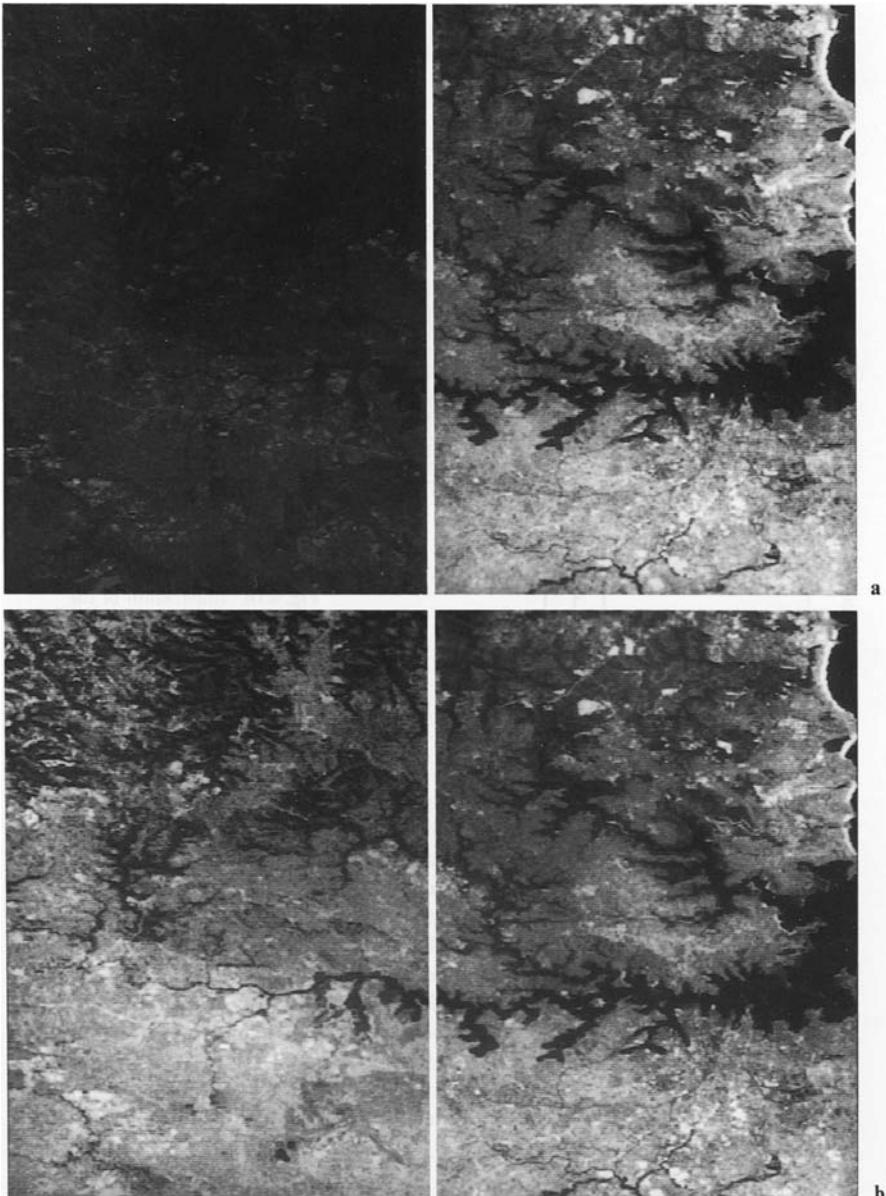
note that some of the new brightness values produced may not be in the available range; as before, they are adjusted to the nearest acceptable value.

An example using a pair of contiguous image segments is shown in Fig. 4.14. Because of seasonal differences the contrasts are quite different. Using the cumulative histograms an acceptable matching is achieved. Such a process, as noted earlier, is an essential step in producing a mosaic of separate contiguous images. Another step is to ensure geometric integrity of the join. This is done using the geometric registration procedures of Sect. 2.5.

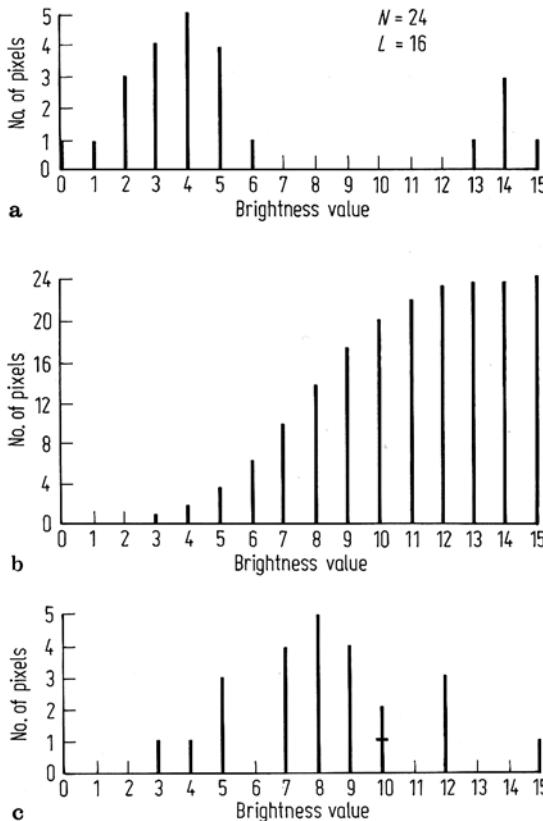
#### 4.5.3

#### Matching to a Mathematical Reference

In some applications it is of value to pre-specify the desired shape of an image histogram to give a modified image with a particular distribution of brightness values. To implement this it is necessary to take an existing image histogram and modify it according to the procedures of Sect. 4.5.1. The reference is a mathematical function that describes the desired shape. A particular example is to match an image histogram to a Gaussian or normal shape. Often this is referred to as applying a “gaussian stretch” to an image; it yields a modified version with few black and white regions and in which most detail is contained in the mid-grey range. This requires a reference histogram in the form of a normal distribution. However since a cumulative version of the reference is to be used, it is really a cumulative normal distribution that is required. Fortunately cumulative normal tables and curves are readily available. To use such a table in the contrast matching situation requires its ordinate to be adjusted to the total number of pixels in the image to be modified and its abscissa to be chosen to match the maximum allowable brightness range in the image. The latter requires consideration to be given to the number of standard deviations of the Gaussian distribution to be contained in the total brightness value range, having in mind that the Gaussian function is continuous to  $\pm\infty$ . The mean of the distribution is placed



**Fig. 4.14.** **a** Contiguous Landsat multispectral scanner images showing contrast and brightness differences resulting from seasonal effects. The left hand image is an autumn scene and that on the right a summer scene, both of the northern suburbs of Sydney, Australia. **b** The same image pair but in which the histogram of the autumn scene has been matched to that of the summer scene



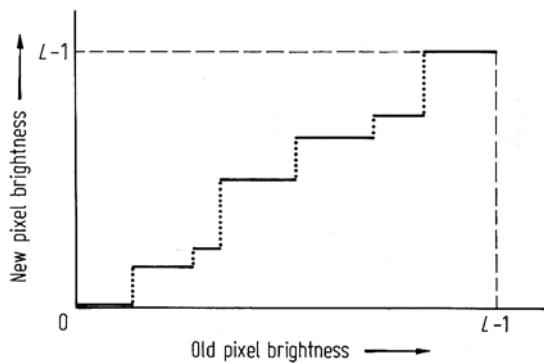
**Fig. 4.15.** Illustration of the modification of an image histogram to a pseudo-Gaussian shape.  
**a** Original histogram; **b** Cumulative normal histogram; **c** Histogram matched to Gaussian reference

usually at the mid-point of the brightness scale and commonly the standard deviation is chosen such that the extreme black and white regions are three standard deviations from the mean. A simple illustration is shown in Fig. 4.15.

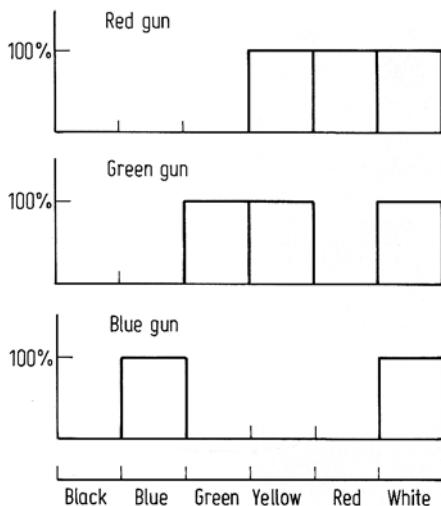
## 4.6 Density Slicing

### 4.6.1 Black and White Density Slicing

A point operation often performed with remote sensing image data is to map *ranges* of brightness value to particular shades of grey. In this way the overall discrete number of brightness values used in the image is reduced and some detail is lost. However the effect of noise can also be reduced and the image becomes segmented,

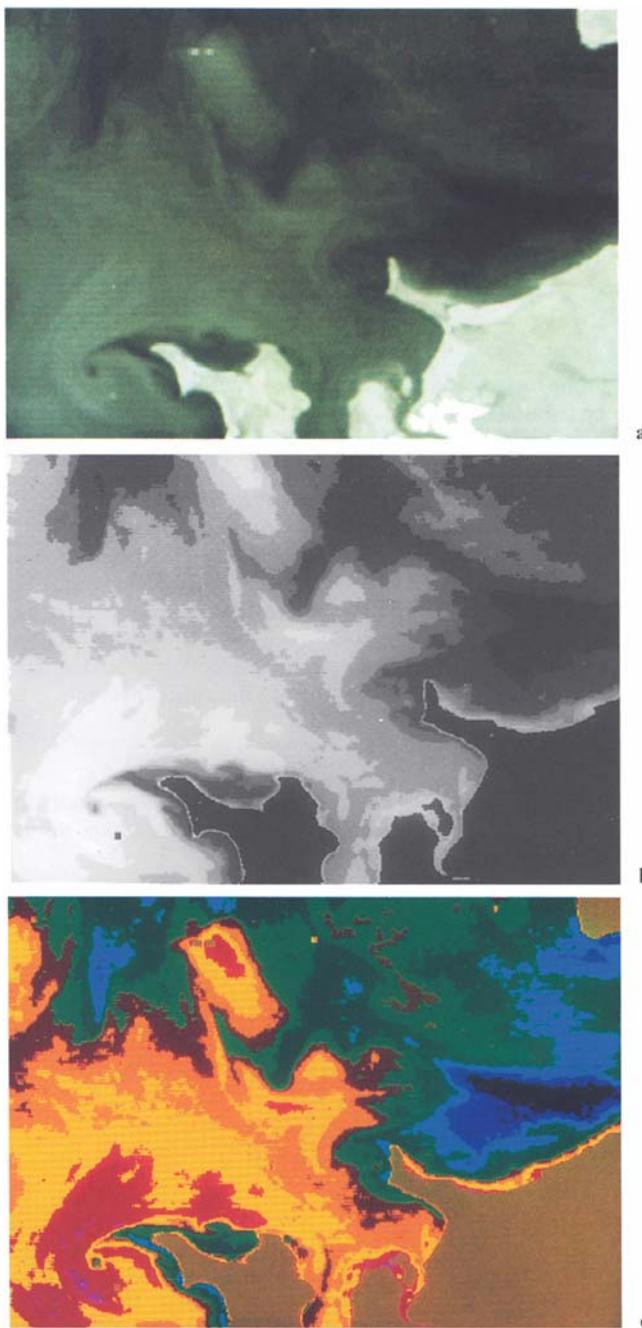


**Fig. 4.16.** The brightness value mapping function corresponding to black and white density slicing. The thresholds are user specified



**Fig. 4.17.** Simple example of creating the look-up tables for a colour display device to implement colour density slicing. Here only six colours have been chosen for simplicity

or sometimes contoured, in sections of similar grey level, in which each segment is represented by a user specified brightness. The technique is known as density slicing and finds value, for example, in highlighting bathymetry in images of water regions when penetration is acceptable. When used generally to segment a scalar image into significant regions of interest it is acting as a simple one dimensional parallelepiped classifier (see Sect. 8.4). The brightness value mapping function for density slicing is as illustrated in Fig. 4.16. The thresholds in such a function are entered by the user. An image in which the technique has been used to highlight bathymetry is shown in Fig. 4.18. Here differences in Landsat multispectral scanner visible imagery, at brightnesses too low to be discriminated by eye, have been mapped to new grey levels to make the detail apparent.



**Fig. 4.18.** Illustration of contouring in water detail using density slicing. **a** The image used is a band 5 + band 7 composite Landsat multispectral scanner image, smoothed to reduce line striping and then density sliced; **b** Black and white density slicing; **c** Colour density slicing

### 4.6.2

#### Colour Density Slicing and Pseudocolouring

A simple yet lucid extension of black and white density slicing is to use colours to highlight brightness value ranges, rather than simple grey levels. This is known as colour density slicing. Provided the colours are chosen suitably, it can allow fine detail to be made immediately apparent. It is a particularly simple operation to implement on a display system by establishing three brightness value mapping functions in the manner depicted in Fig. 4.17. Here one function is applied to each of the colour primaries used in the display device. An example of the use of colour density slicing, again for bathymetric purposes, is given in Fig. 4.18.

This technique is also used to give a colour rendition to black and white imagery. It is then usually called pseudocolouring. Where possible this uses as many distinct hues as there are brightness values in the image. In this way the contours introduced by density slicing are avoided. Moreover it is of value in perception if the hues used are graded continuously. For example, starting with black, moving from dark blue, mid blue, light blue, dark green, etc. through to oranges and reds will give a much more acceptable pseudocoloured product than one in which the hues are chosen arbitrarily.

### References for Chapter 4

Much of the material on contrast enhancement and contrast matching treated in this chapter will be found also in Castleman (1996) and Gonzalez and Woods (1992) but in more mathematical detail. Passing coverages are also given by Moik (1980) and Hord (1982). More comprehensive treatments will be found in Schowengerdt (1997), Jensen (1986), Mather (1987) and Harrison and Jupp (1990).

The papers by A. Schwartz (1976) and J.M. Soha et al. (1976) give examples of the effect of histogram equalization and of Gaussian contrast stretching. Chavez et al. (1979) have demonstrated the performance of multicycle contrast enhancement, in which the brightness value mapping function  $y = f(x)$  is cyclic. Here, several sub-ranges of input brightness value  $x$  are each mapped to the full range of output brightness value  $y$ . While this destroys the radiometric calibration of an image it can be of value in enhancing structural detail.

K.R. Castleman, 1996: Digital Image Processing, 2e, N.J., Prentice-Hall.

P.S. Chavez, G.L. Berlin, and W.B. Mitchell, 1979: Computer Enhancement Techniques of Landsat MSS Digital Images for Land Use/Land Cover Assessment. Private Communication, US Geological Survey, Flagstaff, Arizona.

R.C. Gonzalez and R.E. Woods, 1992: Digital Image Processing, Mass., Addison-Wesley.

B.A. Harrison and D.L.B. Jupp, 1990: Introduction to Image Processing, Canberra, CSIRO.

A. Hogan, 1981: A Piecewise Linear Contrast Stretch Algorithm Suitable for Batch Landsat Image Processing. Proc. 2nd Australasian Conf. on Remote Sensing, Canberra, 6.4.1–6.4.4.

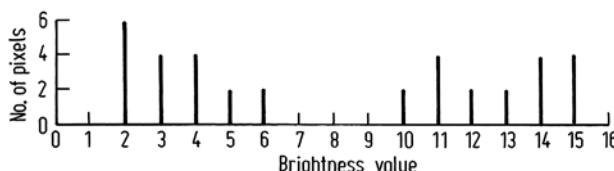
R.M. Hord, 1982: Digital Image Processing of Remotely Sensed Data, N.Y., Academic.

J.R. Jensen, 1986: Introductory Digital Image Processing — a Remote Sensing Perspective. N.J., Prentice-Hall.

- P.M. Mather, 1987: Computer Processing of Remotely-Sensed Images. Suffolk, Wiley.
- J.G. Moik, 1980: Digital Processing of Remotely Sensed Images, Washington, NASA.
- A. Schwartz, 1976: New Techniques for Digital Image Enhancement, in Proc. Caltech/JPL Conf. on Image Processing Technology, Data Sources and Software for Commercial and Scientific Applications, California, Nov. 3–5, 2.1–2.12.
- R.A. Schowengerdt, 1997: Remote Sensing Models and Methods for Image Processing, 2e, New York, Academic.
- J.M. Soha, A.R. Gillespie, M.J. Abrams and D.P. Madura, 1976: Computer Techniques for Geological Applications; in Proc. Caltech/JPL Conf. on Image Processing Technology, Data Sources and Software for Commercial and Scientific Applications, Nov. 3–5, 4.1–4.21.

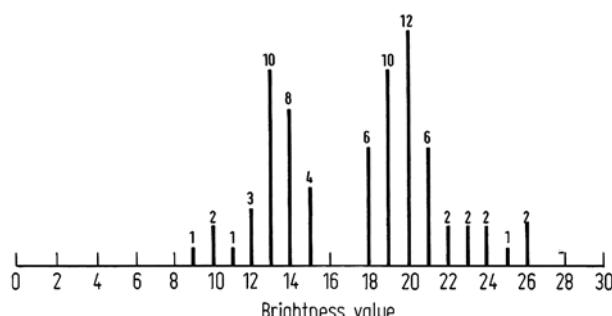
## Problems

- 4.1** One form of histogram modification is to match the histogram of an image to a Gaussian or normal function. Suppose a raw image has the histogram indicated in Fig. 4.19. Produce the look-up table that describes how the brightness values of the image should be changed if the histogram is to be mapped, as nearly as possible, to a Gaussian histogram with a mean of 8 and a standard deviation of 2 brightness values. *Note that the sum of counts in the Gaussian reference histogram must be the same as that in the raw data histogram.*



**Fig. 4.19.** Histogram

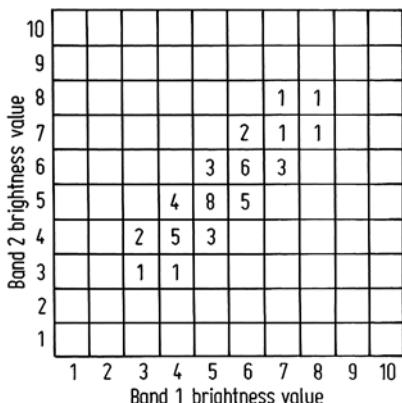
- 4.2** The histogram of a particular image is shown in Fig. 4.20. Produce the modified version that results from:



**Fig. 4.20.** Histogram of a single dimensional image

- (i) a simple linear contrast stretch which makes use of the full range of brightness values
- (ii) a simple piecewise linear stretch that maps the range (12, 23) to (0, 31) and
- (iii) histogram equalization (i.e. producing a quasi-uniform histogram).

**4.3** A two-dimensional histogram for particular two band image data is shown in Fig. 4.21. Determine the histogram that results from a simple linear contrast stretch on each band individually.



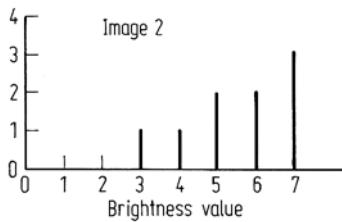
**Fig. 4.21.** Two-dimensional histogram

**4.4** Determine, algebraically, the contrast mapping function that equalizes the contrast of an image which has a Gaussian histogram at the centre of the brightness value range, with the extremities of the range being three standard deviations from the mean.

**4.5** What is the shape of the cumulative histogram of an image that has been contrast (histogram) equalized? Can this be used as a figure of merit in histogram equalization?

**4.6** Clouds and large regions of clear, deep water frequently give histograms for near infrared imagery that have large high brightness level or low brightness value bars respectively. Sketch histograms of these types. Qualitatively, equalize the histograms using the material of Sect. 4.4 and comment on the undesirable appearance of the corresponding contrast enhanced images. Show that the situation can be rectified somewhat by artificially limiting the large bars to values not greatly different to the heights of other bars in the histogram, provided the accompanying cumulative histograms are normalised to correspond to the correct number of pixels in the image. A similar, but more effective procedure has been given in A. Hogan (1981).

**4.7** Two Landsat images are to be joined side by side to form a mosaic for a particular application. To give the new, combined image a uniform appearance it is decided that the range and distribution of brightness levels in the first image should be made to match those of the second image, before they are joined. This is to be carried out by matching the histogram of image 1 to that of image 2. The original histograms are shown in Fig. 4.22. Produce a look up table that can be used to transform the pixel brightness values of image 1 in order to match the histograms as nearly as possible. Use the look-up table to modify the histogram of image 1 and comment on the degree to which contrast matching has been achieved.

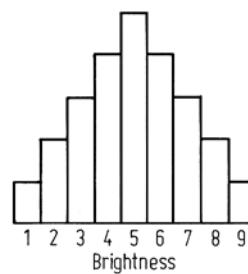
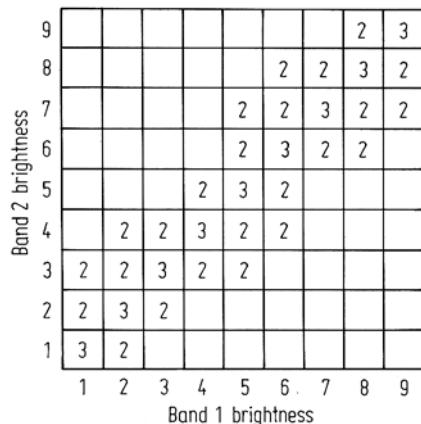


**Fig. 4.22.** Histograms of image 1 and image 2

**4.8** (a) Contrast enhancement is frequently carried out on remote sensing image data. Describe the advantages in doing so, if the data is to be analysed by

- (i) photointerpretation
- (ii) quantitative computer methods.

(b) A particular two band image has the two dimensional histogram shown in Fig. 4.23. It is proposed to enhance the contrast of the image by matching the histograms in each band to the triangular profile shown. Produce look-up tables to enable each band to be enhanced, and from these produce the new two-dimensional histogram for the image.



**Fig. 4.23.** Two dimensional histogram

**4.9** Plot the equilized histogram for the example of Table 4.2. Compare it with Fig. 4.9 and comment on the effect of restricting the range of output brightnesses. Repeat the exercise for the cases of 4 and 2 output brightness values.

**4.10** Suppose a particular image has been modified by (i) linear contrast enhancement and (ii) by histogram equalisation. Suppose you have available the digital image data for both the original image and the contrast modified versions. By inspecting the data (or histograms) describe how you would determine quantitatively which technique was used in each case.

# 5

## Geometric Enhancement Using Image Domain Techniques

### 5.1

#### Neighbourhood Operations

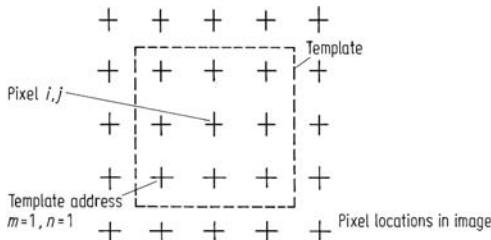
This chapter presents methods by which the geometric detail in an image may be modified and enhanced. The specific techniques covered are applied to the image data directly and could be called image domain techniques. These are alternatives to procedures used in the spatial frequency domain which require Fourier transformation of the image beforehand. Those are treated in Chap. 7.

In contrast to the point operations used for radiometric enhancement, techniques for geometric enhancement are characterised by operations over neighbourhoods. The procedures still determine modified brightness values for an image's pixels; however, the new value for a given pixel is derived from the brightnesses of a set of the surrounding pixels. It is this spatial interdependence of the pixel values that leads to variations in the perceived image geometric detail. The neighbourhood influence will be apparent readily in the techniques of this chapter; for the Fourier transformation methods of Chap. 7 it will be discerned in the definition of the Fourier operation.

### 5.2

#### Template Operators

Geometric enhancements of most interest in remote sensing generally relate to smoothing, edge detection and enhancement, and line detection. Enhancement of edges and lines leads to image sharpening. Each of these operations is considered in the following sections. Most of the methods to be presented are, or can be expressed as, template techniques in which a template, box or window is defined and then moved over the image row by row and column by column. The products of the pixel brightness values, covered by the template at a particular position, and the template entries, are taken and summed to give the template response. This response is then used to define a new brightness value for the pixel currently at the centre of the template. When this is done for every pixel in the image, a radiometrically modified



**Fig. 5.1.** A  $3 \times 3$  template positioned over a group of nine image pixels, showing the relative locations of pixels and template entry addresses

image is produced that enhances or smooths geometric features according to the specific numbers loaded into the template. A  $3 \times 3$  template is illustrated in Fig. 5.1. Templates of any size can be defined, and for an  $M$  by  $N$  pixel sized template, the response for image pixel  $i, j$  is

$$r(i, j) = \sum_{m=1}^M \sum_{n=1}^N \phi(m, n) t(m, n) \quad (5.1)$$

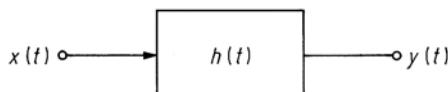
where  $\phi(m, n)$  is the pixel brightness value, addressed according to the template position and  $t(m, n)$  is the template entry at that location. Often the template entries collectively are referred to as the ‘kernel’ of the template and the template technique generally is called convolution, in view of its similarity to time domain convolution in linear system theory. This concept is developed in Sect. 5.3 below.

## 5.3

### Geometric Enhancement as a Convolution Operation

This section presents a brief linear system theory basis for the use of the template expression of (5.1). It contains no results essential to the remainder of the chapter and can be safely passed over by the reader satisfied with (5.1) from an intuitive viewpoint.

Consider a signal in time represented as  $x(t)$ . Suppose this is passed through a system of some sort to produce a modified signal  $y(t)$  as depicted in Fig. 5.2. The system here could be an intentional one such as an amplifier or filter, inserted to change the signal in a predetermined way; alternatively it could represent unintentional modification of the signal such as by distortion or the effect of noise. The properties of the system can be described by a function of time  $h(t)$ . This is called



**Fig. 5.2.** Signal model of a linear system

its impulse response (or sometimes its transfer function, although that term is more properly used for the Fourier transform of the impulse response, as noted in Chap. 7).

The relationship between  $y(t)$  and  $x(t)$  is described by the convolution operation. This can be expressed as an integral

$$y(t) = \int_{-\infty}^{\infty} x(\tau) h(t - \tau) d\tau \triangleq x(t) * h(t) \quad (5.2)$$

as shown in McGillem and Cooper (1984). McGillem and Cooper, Castleman (1996) and Brigham (1974, 1988) all give comprehensive accounts of the properties of convolution and the characteristics of linear systems derived from the operation of convolution.

A similar mathematical description applies when images are used in place of signals in (5.2) and Fig. 5.2. The major difference is that the image has two independent variables (its  $i$  and  $j$  pixel position indices, or address) whereas the signal  $x(t)$  in Fig. 5.2 has only one – time. Consequently the transfer function of a system that operates on an image is also two dimensional, and the processed image is given by a two dimensional version of the convolution integral in (5.2). In this case the system can represent any process that modifies the image. It could, for example, account for degradation brought about by the finite point spread function of an image acquisition instrument or an image display device. It could also represent the effect of intentional image processing such as that used in geometric enhancement. In both cases if the new and old versions of the image are described by  $r(x, y)$  and  $\phi(x, y)$  respectively, where  $x$  and  $y$  are continuous position variables that describe the locations of points in a continuous image domain, then the two dimensional convolution operation is described as

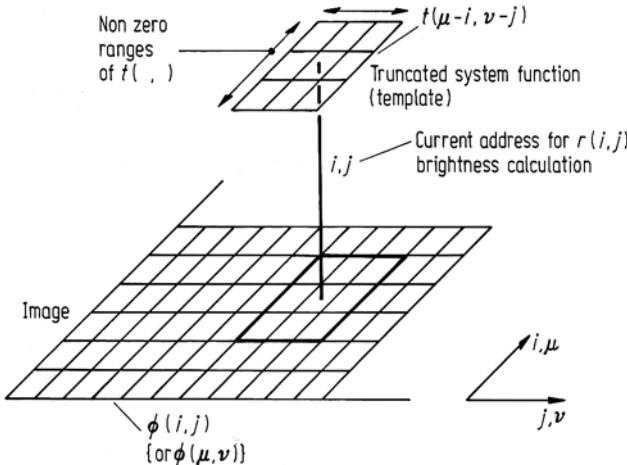
$$r(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \phi(u, v) t'(x - u, y - v) du dv \quad (5.3)$$

where  $t'(x, y)$  is the two dimensional system transfer function (impulse response). It will also be called the system function here.

Even though, in principle,  $\phi(x, y)$  and  $t'(x, y)$  are both defined over the complete range of  $x$  and  $y$ , in practice they are both limited. Clearly the image itself must be finite in extent spatially; the system function  $t'(x, y)$  is also generally quite limited. Should it represent the point spread function of an imaging device it would be significantly non-zero over only a small range of  $x$  and  $y$ . (If it were an impulse it can be shown that (5.3) yields  $r(x, y) = \phi(x, y)$  as would be expected).

In order to be applicable to digital image data it is necessary to modify (5.3) so that the discrete natures of  $x$  and  $y$  are made explicit and, consequently, the integrals are replaced by suitable summations. If we let  $i, j$  represent discrete values of  $x, y$  and similarly  $\mu, v$  represent discrete values of the integration variables  $u, v$  then (5.3) can be written

$$r(i, j) = \sum_{\mu} \sum_{v} \phi(\mu, v) t'(i - \mu, j - v) \quad (5.4)$$



**Fig. 5.3.** An illustration of the operations implicit in (5.4)

which is a digital form of the two dimensional convolution integral. The sums are taken over all values of  $\mu, \nu$  for which a non-zero result exists.

To see how (5.4) would be used in practice it is necessary to interpret the sequence of operations implied. For clarity, assume the non-zero range of  $t'(i, j)$  is quite small compared with that for the image  $\phi(i, j)$ . Also assume  $t'(i, j)$  is a square array of samples, for example  $3 \times 3$ . These assumptions in no way prejudice the generality of what follows.

In (5.4) the negative signs on  $\mu$  and  $\nu$  in  $t'(i - \mu, j - \nu)$  imply a reflection through both axes. This is tantamount to a rotation of the system function through  $180^\circ$  before any further operations take place. Let the rotated form be called  $t(\mu - i, \nu - j)$ .

Equation (5.4) implies that a brightness value for the response image at pixel location  $i, j$  -viz.  $r(i, j)$  is given by taking the non-zero products of the original version of the image and the rotated system function and adding these together. In so doing, note that the origin of the  $\mu, \nu$  co-ordinates is the same as for the  $i, j$  co-ordinates just as the dummy and real variable co-ordinates in (5.2) and (5.3) are the same. Also note that the effect of  $\mu - i, \nu - j$  in  $t(\mu - i, \nu - j)$  is to shift the origin of the rotated system function to the location  $i, j$  – the current pixel address for which a new brightness value is to be calculated. These two points are illustrated in Fig. 5.3. The need to produce brightness values for pixels in the response image at every  $i, j$  means that the origin of the rotated system function must be moved progressively, implying that a different set of products between the original image and rotated system function is taken every time.

The sequence of operations described between the rotated system function and the original image are the same as those noted in Sect. 5.2 in regard to (5.1). The only difference in fact between (5.1) and (5.4) lies in the definitions of the indices  $m, n$  and  $\mu, \nu$ . In (5.1) the pixel addresses are referred to an origin defined at the bottom left hand corner of the template, with the successive shifts mentioned in the

accompanying description. This is a simple way to describe the template and readily allows any template size to be defined. In (5.4) the shifts are incorporated into the expression by defining the image and system function origins correctly.

The templates of Sect. 5.2 are equivalent to the rotated system functions of this section. Consequently any image modification operation that can be modelled by convolution, and described in principle in a manner similar to that in Fig. 5.2, can also be expressed in template form. For example, if the point spread function of a display device is known, then an equivalent template can be devised, noting that the 180° rotation is important if the system function is not symmetric. In a like manner intentional modifications of an image – such as smoothing and sharpening – can also be implemented using templates. The actual template entries to be used can often be developed intuitively, having careful regard to the desired results. Alternatively the system function  $t'(i, j)$  necessary to implement a particular desired filtering operation can be defined first in the spatial frequency domain, using the material from Chap. 7, and then transformed back to the image domain. Rotation by 180° then gives the required template.

## 5.4

### Image Domain Versus Fourier Transformation Approaches

Most geometric enhancement procedures can be implemented using either the Fourier transform approach of Chap. 7 or the image domain procedures of this chapter. Which option to use depends upon several factors such as available software, familiarity with each method including its limitations and idiosyncrasies, and ease of use. A further consideration relates to computer processing time. This last issue is pursued here in order to indicate, from a cost viewpoint, when one method should be chosen in favour of the other.

Both the Fourier transform, frequency domain process and the template approach consist only of sets of multiplications and additions. No other mathematical operations are involved. It is sufficient, therefore, from the point of view of cost, to make a comparison based upon the number of multiplications and number of additions necessary to achieve a result. Here we will ignore the additions since they are generally faster than multiplications for most computers and also since they are comparable in number to the multiplications involved.

For an image of  $K \times K$  pixels (only a square image is considered for simplicity) and a template of size  $M \times N$  the total number of multiplications necessary to evaluate (5.1) for every image pixel (ignoring any difficulties with the edges of the image) is

$$N_C = MNK^2 \quad (5.5a)$$

From the material presented in Sect. 7.8.4 it can be seen that the number of (complex) multiplications required in the frequency domain approach is,

$$N_F = 2K^2 \log_2 K + K^2 \quad (5.5b)$$

**Table 5.1.** Time comparison of geometric enhancement by template operation compared with the Fourier transformation approach – based upon multiplication count comparison, described by (5.6) in which the added cost of complex multiplication is ignored

Image size	Template size	$3 \times 3$	$3 \times 5$	$5 \times 5$	$5 \times 7$	$7 \times 7$
$16 \times 16$		1.00	1.67	2.78	3.89	5.44
$64 \times 64$		0.69	1.15	1.92	2.69	3.77
$128 \times 128$		0.60	1.00	1.67	2.33	3.27
$256 \times 256$		0.53	0.88	1.47	2.06	2.88
$512 \times 512$		0.47	0.79	1.32	1.84	2.58
$1024 \times 1024$		0.43	0.71	1.19	1.67	2.33
$2048 \times 2048$		0.39	0.65	1.09	1.52	2.13
$4096 \times 4096$		0.36	0.60	1.00	1.40	1.96

A cost comparison therefore is

$$\frac{N_C}{N_F} = MN/(2\log_2 K + 1) \quad (5.6)$$

When this figure is less than unity it is more economical to use the template operator approach. Otherwise the Fourier transformation procedure is more cost-effective. Clearly this does not take into account program overheads (such as the bit shuffling required in the frequency domain approach, how data is buffered into computer memory from disc for processing) and the added cost of complex multiplications; however it is a reasonable starting point in choosing between the methods.

Table 5.1 contains a number of values of  $N_C/N_F$  for various image and template sizes, from which it is seen that, provided a  $3 \times 3$  template will implement the enhancement required, then it is always more cost-effective than enhancement based upon Fourier transformation. Similarly, a non isotropic  $3 \times 5$  template is more cost-effective for practical image sizes. However the spatial frequency domain technique will be economical if very large templates are needed, although only marginally so for large images.

As a final comment in this comparison it should be remarked that the frequency domain method is able to implement processes not possible (or at least not viable) with template operators. Removal of periodic noise is one example. This is particularly simple in the spatial frequency domain but requires unduly complex templates or even nonlinear operators (such as median filtering) in the image domain. Notwithstanding these remarks the template approach is a popular one since often  $3 \times 3$  and  $5 \times 5$  templates are sufficient to achieve desired results.

## 5.5

# Image Smoothing (Low Pass Filtering)

### 5.5.1

## Mean Value Smoothing

Images can contain random noise superimposed on the pixel brightness values owing to noise generated in the sensors that acquire the image data, systematic quantisation noise in the signal digitising electronics and noise added to the video signal during transmission. This will show as a speckled ‘salt and pepper’ pattern on the image in regions of homogeneity; it can be removed by the process of low pass filtering or smoothing, unfortunately usually at the expense of some high frequency information in the image. To smooth an image a uniform template in (5.1) is used with entries

$$t(m, n) = 1/MN \quad \text{for all } m, n$$

so that the template response is a simple average of the pixel brightness values currently within the template, viz

$$r(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \phi(m, n) \quad (5.7)$$

The pixel at the centre of the template is thus represented by the average brightness level in a neighbourhood defined by the template dimensions. This is an intuitively obvious template for smoothing and is equivalent to using running averages for smoothing time series information.

It is evident that high frequency information such as edges will also be averaged and lost. This loss of high frequency detail can be circumvented somewhat if a threshold is applied to the template response in the following manner,

Let

$$\varrho(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N \phi(m, n)$$

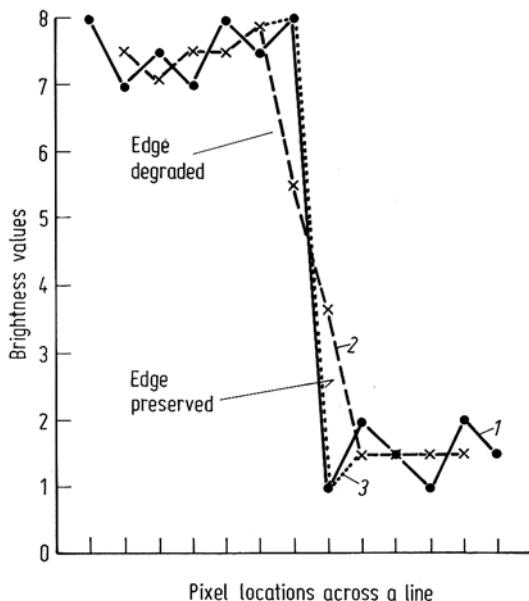
then

$$\begin{aligned} r(i, j) &= \varrho(i, j) \quad \text{if } |\phi(i, j) - \varrho(i, j)| < T \\ &= \phi(i, j) \quad \text{otherwise} \end{aligned}$$

where  $T$  is a prespecified threshold.  $T$  could be determined *a priori* based upon knowledge of or an estimate of scene signal to noise ratio.

Eliason and McEwan (1990) recommend choosing the threshold as a multiple of the standard deviation of brightness within the template window. This provides better noise removal in homogeneous regions while allowing better preservation of edges and other valid high spatial frequency detail.

A simple illustration of image smoothing by averaging over a template, both with and without the application of a threshold, is given in Fig. 5.4. For clarity this is based upon a hypothetical one dimensional image, or alternatively a single line of



**Fig. 5.4.** Illustration of the effect of  $3 \times 1$  averaging across a single line of image data with and without thresholding. Note, thresholding preserves edges while reducing noise. 1 original image, 2  $3 \times 1$  smoothing, 3  $3 \times 1$  smoothing with threshold of 1

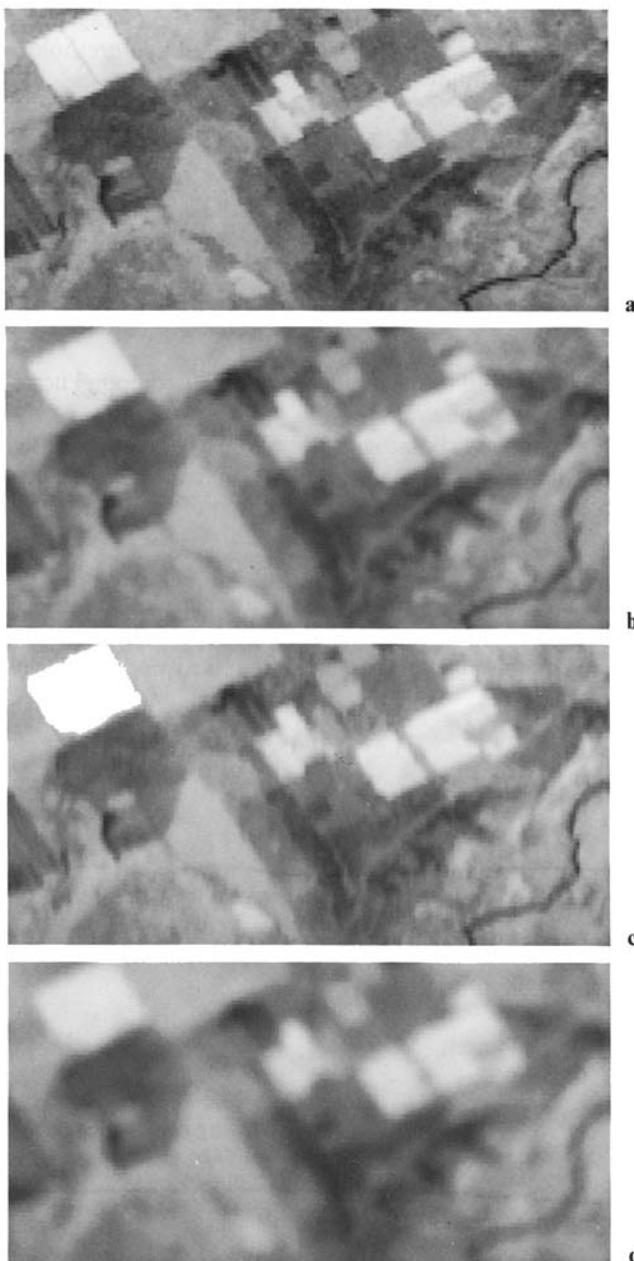
image data, with which a  $3 \times 1$  template is used. In this manner the actual numerical modification of pixel brightness values can be observed,

In principle, templates of any shape and size can be used. Larger templates give more smoothing (and greater loss of high frequency detail) whereas horizontal rectangular templates will smooth horizontal noise but leave noise and high frequency detail in the vertical direction relatively unaffected by comparison. In Fig. 5.5 several different smoothing templates have been applied to a Landsat multispectral scanner infrared image.

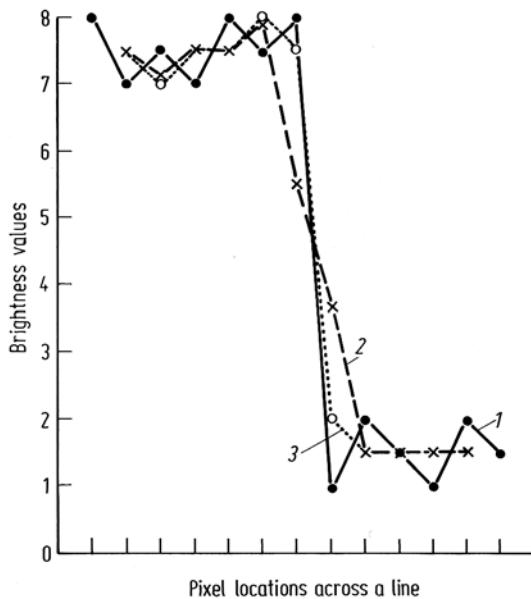
Commonly, smoothing by template methods is referred to as box car filtering. When based upon (5.7) it is also called mean value smoothing, or averaging.

### 5.5.2 Median Filtering

Disadvantages of the thresholding method for avoiding edge deterioration are that it adds to the computational cost of the smoothing operation and  $T$  must be determined. An alternative technique for smoothing in which the edges in an image are maintained is that of median filtering. In this the pixel at the centre of the template is given the median brightness value of all the pixels covered by the template – i. e. that value which has as many values higher and lower. (For example, the median of 4, 6, 3, 7, 9, 2, 1, 8, 8 is 6, whereas the mean is 5.3). Figure 5.6 shows the effect of median



**Fig. 5.5.** Examples of mean value smoothing of a Landsat multispectral scanner infrared (band 7) image. **a** Original; **b**  $3 \times 3$  smoothed version; **c**  $3 \times 1$  smoothed version; **d**  $5 \times 5$  smoothed version



**Fig. 5.6.** Comparison of simple averaging and median filtering of a single line of image data.  
1 original image, 2  $3 \times 1$  smoothing, 3  $3 \times 1$  median filtering

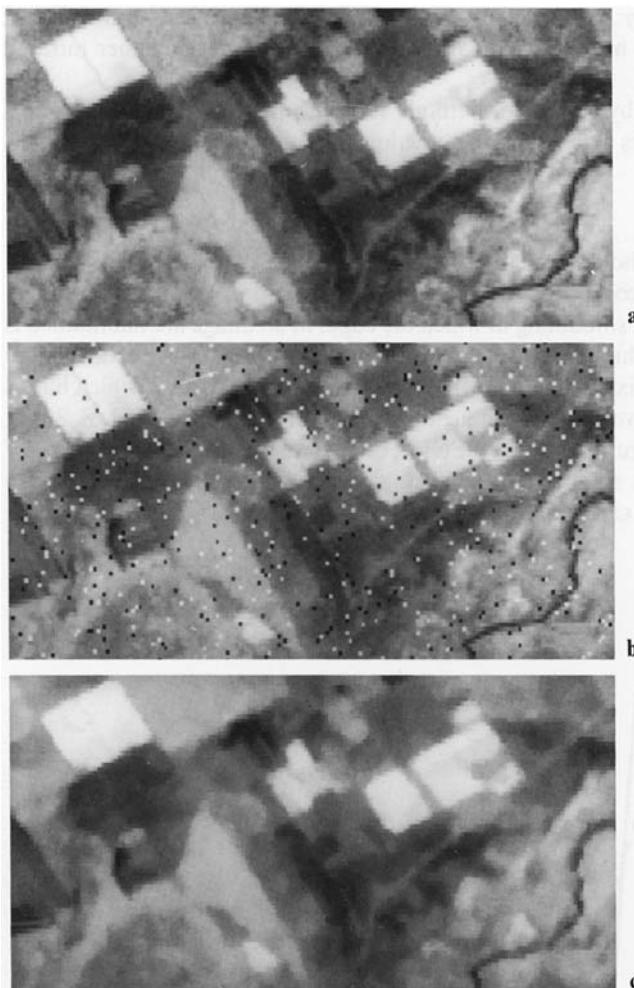
filtering on a single line of image data compared with simple box car averaging, which uses the mean of pixel brightness values. Again, it can be seen that most of the original edge is preserved.

An application for which median filtering is well suited is the removal of impulse-like noise. This is because pixels corresponding to noise spikes are atypical in their neighbourhood and will be replaced by the most typical pixel in that neighbourhood. Figure 5.7 gives an example of median filtering on an image with added black and white impulsive noise.

Finally it should be noted that median filtering is not a linear function of the brightness values of the image pixels. Consequently it is not a convolution operation in the sense described in Sect. 5.3.

## 5.6 Edge Detection and Enhancement

Edge enhancement is a particularly simple and effective means for increasing geometric detail in an image. It is performed by first detecting edges and then either adding these back into the original image to increase contrast in the vicinity of an edge, or highlighting edges using saturated (black, white or colour) overlays on borders.



**Fig. 5.7.** Illustration of the effect of median filtering on an image which contains impulsive noise. **a** Original image; **b** Image with noise; **c** Filtered image

There are essentially three economical techniques for detecting edges using image domain techniques. They are

- (i) by using an edge detecting template,
- (ii) by calculating spatial derivatives, or
- (iii) by subtracting a smoothed image from its original.

These three approaches are treated in the following sections.

### 5.6.1

#### Linear Edge Detecting Templates

A  $3 \times 3$  template that detects vertical edges in image data is

$$t(m, n) = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad (5.8a)$$

As can be inferred from its structure it computes a value for the central pixel under the template that is the accumulated difference horizontally between pixels on three adjacent rows. To see this, consider a region of an image which is basically dull (brightness value 2) into which protrudes a bright object (brightness value 8) as depicted in Fig. 5.8a. Application of the template yields the responses shown in Fig. 5.8b, in which the vertical edge between the object and background has been detected but not the horizontal edge. Note that the edge is defined by two columns of pixels, one on either side of the true edge position. A threshold would normally be applied to the template response (say 9 in the case of Fig. 5.8) to define the edge pixels.

Templates for detecting edges in other orientations are:

$$\begin{array}{ccc} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ +1 & +1 & +1 \end{bmatrix} & \begin{bmatrix} 0 & +1 & +1 \\ -1 & 0 & +1 \\ -1 & -1 & 0 \end{bmatrix} & \begin{bmatrix} +1 & +1 & 0 \\ +1 & 0 & -1 \\ 0 & -1 & -1 \end{bmatrix} \\ \text{horizontal} & & \text{diagonal} \end{array} \quad (5.8b)$$

Clearly all four  $3 \times 3$  templates have to be applied to an image to detect its edges in all orientations. This requires four passes over the image data, computing each template response for each pixel. At the completion of all processing the four template responses for each pixel are compared and the pixel labelled (as an edge in a particular direction) according to the largest template response provided that the response is

2 2 2 2 8 8 8 8	.	.	.	.	.	.
2 2 2 2 8 8 8 8	.	0 0	18 18	0 0	.	.
2 2 2 2 8 8 8 8	.	0 0	18 18	0 0	.	.
2 2 2 2 8 8 8 8	.	0 0	12 12	0 0	.	.
2 2 2 2 2 2 2 2	.	0 0	6 6	0 0	.	.
2 2 2 2 2 2 2 2	.	0 0	0 0	0 0	0 0	.
2 2 2 2 2 2 2 2	.	0 0	0 0	0 0	0 0	.
dull background	.	.	.	.	.	.

**a**

vertical edge 2 pixels wide  
**b**

**Fig. 5.8.** Image **a** and edges detected by a vertically sensitive template **b**; Dots indicate indeterminate edge responses for this example

also above a user specified threshold. Choosing a threshold too low will lead to many false edge counts. These contribute to noise in the processed image. Conversely, if the threshold is set too high, there will be little continuity in the detected edges.

### 5.6.2 Spatial Derivative Techniques

If an image consists of a continuous brightness function of a pair of continuous coordinates,  $x$  and  $y$ , say  $\phi(x, y)$ , then a vector gradient can be defined in the image according to

$$\nabla\phi(x, y) = \frac{\partial}{\partial x}\phi(x, y)\mathbf{i} + \frac{\partial}{\partial y}\phi(x, y)\mathbf{j} \quad (5.9)$$

where  $\mathbf{i}, \mathbf{j}$  are a pair of unit vectors. The direction of the vector gradient is the direction of maximum upward slope and its amplitude is the value of the slope. For edge detection operations usually only the magnitude of the gradient, defined by

$$|\nabla| = \sqrt{\nabla_1^2 + \nabla_2^2} \quad (5.10a)$$

is retained, in which

$$\nabla_1 = \frac{\partial}{\partial x}\phi(x, y) \quad \nabla_2 = \frac{\partial}{\partial y}\phi(x, y) \quad (5.10b)$$

The direction of the gradient is usually of interest only in contouring applications or in determining aspect in digital terrain models.

#### 5.6.2.1 The Roberts Operator

For digital image data, in which  $x$  and  $y$  are discrete, the continuous derivatives in (5.10) are replaced by differences. For example, it is possible to define

$$\nabla_1 = \phi(i, j) - \phi(i + 1, j + 1) \quad (5.11a)$$

and

$$\nabla_2 = \phi(i + 1, j) - \phi(i, j + 1) \quad (5.11b)$$

which are the discrete components of the vector derivative at the point  $i + \frac{1}{2}, j + \frac{1}{2}$ , in the diagonal directions. This estimate of gradient is called the Roberts operator, and is by definition associated with the pixel  $i, j$ .

Application of the Roberts operator to the model image at Fig. 5.8a yields the results shown in Fig. 5.9a, in which it will be seen that both horizontal and vertical edges are detected, as will be diagonal edges. Since this procedure computes a local gradient it is necessary to choose a threshold value above which edge gradients are said to occur. This is usually chosen with experience of a particular image. Frequently however it is useful to produce gradient maps in which pixels, for which the local gradient lies within prespecified upper and lower bounds, are displayed. Conventionally, the responses are placed to the left and upper sides of the edges.

$$\begin{array}{ccccccccc}
 0 & 0 & 0 & 8.5 & 0 & 0 & 0 & \cdot & \cdot \\
 0 & 0 & 0 & 8.5 & 0 & 0 & 0 & \cdot & 0 & 0 & 24 & 24 & 0 & 0 \\
 0 & 0 & 0 & 8.5 & 0 & 0 & 0 & \cdot & 0 & 0 & 24 & 24 & 0 & 0 \\
 0 & 0 & 0 & 6.0 & 8.5 & 8.5 & 8.5 & \cdot & 0 & 0 & 19 & 25 & 24 & 24 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 8 & 19 & 24 & 24 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdot & 0 & 0 & 0 & 0 & 0 & 0 \\
 \cdot & \cdot
 \end{array}$$

**a****b**

**Fig. 5.9.** Response of **a** the Robert's operator and **b** the Sobel operator to the model image data of Fig. 5.8a. Dots are indeterminate responses from edge pixels

### 5.6.2.2

#### The Sobel Operator

A better edge estimator than the Roberts operator is the Sobel operator, which computes discrete gradient in the horizontal and vertical directions *at* the pixel location  $i, j$ . For this, which is clearly more costly to evaluate, the orthogonal components of gradient are

$$\begin{aligned}
 \nabla_1 = & \{\phi(i-1, j+1) + 2\phi(i-1, j) + \phi(i-1, j-1)\} \\
 & - \{\phi(i+1, j+1) + 2\phi(i+1, j) + \phi(i+1, j-1)\}
 \end{aligned} \quad (5.12a)$$

and

$$\begin{aligned}
 \nabla_2 = & \{\phi(i-1, j+1) + 2\phi(i, j+1) + \phi(i+1, j+1)\} \\
 & - \{\phi(i-1, j-1) + 2\phi(i, j-1) + \phi(i+1, j-1)\}
 \end{aligned} \quad (5.12b)$$

Applying this to the example of Fig. 5.8a produces the responses shown in Fig. 5.9b. Again, both horizontal and vertical edges are detected as will be edges on a diagonal slope. As before, a threshold on the responses is generally chosen to allow an edge map to be produced in which small responses, resulting from noise or minor gradients, are suppressed. Also gradient maps can be produced illustrating regions in which the local slope lies within user specified bounds.

It can be seen that the Sobel operator is equivalent to simultaneous application of the templates:

$$\nabla_1 \equiv \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \nabla_2 \equiv \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

### 5.6.2.3

#### The Prewitt Operator

The template of (5.8a) effectively implements a spatial derivative in the horizontal direction. If its vertical counterpart in (5.8b) is applied as well, and the results

combined in (5.10a), then the magnitude of a spatial derivative is generated. This is referred to as the Prewitt operator.

### 5.6.3

#### Thinning, Linking and Border Responses

Should an edge map be of interest (or indeed a line map using the methods of Sect. 5.7) then the product resulting from using the above procedures is likely to contain many double width, or wider lines, such as those seen in Figs. 5.8 and 5.9 and may have lines with many breaks. Such a map can be tidied up by thinning edges or lines that are too thick and by linking together segments that appear to belong to the same edge but are separated by a break. Thinning and linking are not commonly employed in remote sensing image analysis. However should they require consideration available techniques will be found in Babu and Nevatia (1980), Paul and Shanmugan (1982) and Castleman (1996).

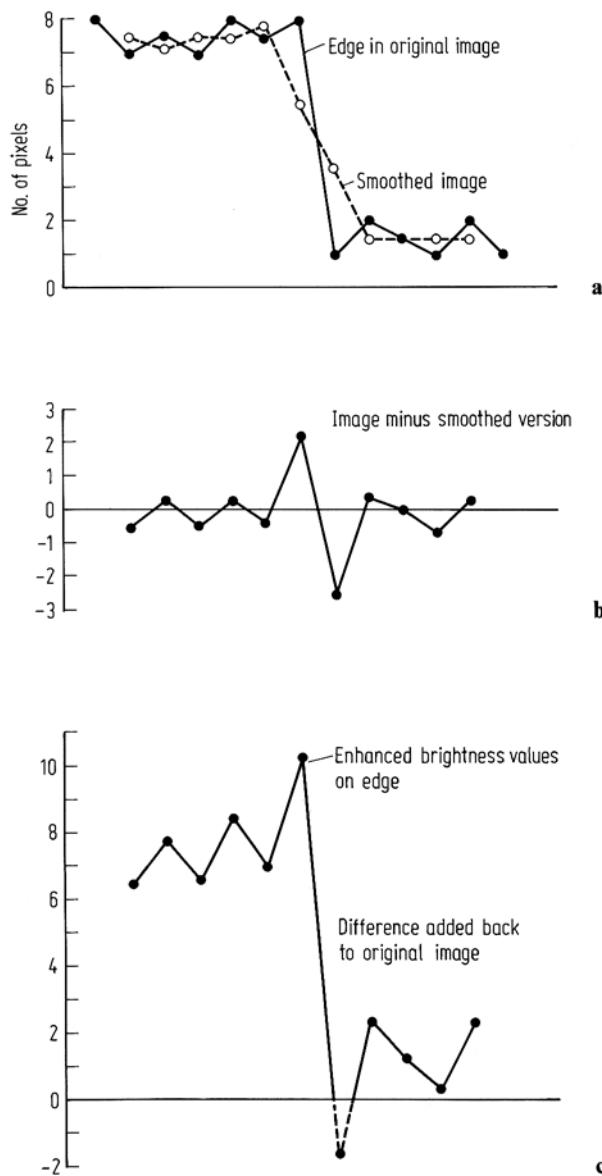
In the examples of Figs. 5.8 and 5.9 border pixels for which detector responses could not be determined were simply left unprocessed. Since images encountered in remote sensing are frequently much larger than  $100 \times 100$  pixels, this is a common practice as the loss of borders is not all that significant. A more elegant means for treating edge pixels however is to create artificial borders of pixels around the image. These are used in the generation of edge pixel responses but are not themselves replaced by a template response. The values given to the artificial border pixels can be taken simply from the adjacent image pixels or, more acceptably from a theoretical viewpoint, they can be taken from the pixels on the extreme opposite edge of the image if only small templates are used. This is based upon the concept, drawn from digital signal processing, that the image, being spatially discretised or sampled, should be regarded as one period both horizontally and vertically of an infinite periodic replication of the array of pixels.

### 5.6.4

#### Edge Enhancement by Subtractive Smoothing (Sharpening)

While treated in the context of edge enhancement this technique really leads to the enhancement of all high spatial frequency detail in an image including edges, lines and points of high gradient. It is probably better regarded therefore as a sharpening technique.

A smoothed image retains all low spatial frequency information but has its high frequency features, such as edges and lines, attenuated (unless edge preservation procedures such as thresholding are employed). Consequently, if a smoothed image is subtracted from its original the resultant difference image will have only the edges and lines substantially remaining. This is illustrated for a single line of image data in Fig. 5.10. After the edges are determined in this manner, the difference image can be added back to the original (in varying proportions) to give an edge enhanced image. This is also illustrated in Fig. 5.10.



**Fig. 5.10.** Edge enhancement by subtractive smoothing. **a** Original line of image data, along with smoothed version; **b** Original line of data minus the smoothed version to leave ‘edges’ detected; **c** Addition of ‘edges’ (general high frequency detail) back to the original image to provide a sharpened version

The difference operation to create a high spatial frequency image can give negative brightness values as seen in Fig. 5.10b. Provided the image is not displayed, this produces no problems. For display however it is common to scale the difference image such that a zero difference is displayed as mid-grey with positive differences towards white and negative differences towards black. When the difference image is added back to the original, negative brightnesses can again result. Again, this can be handled by level shifting or scaling, or simply by setting negative brightness values to zero.

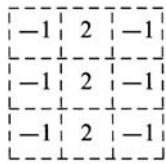
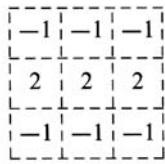
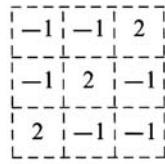
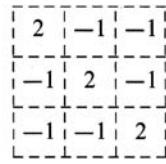
Figure 5.11 shows the sharpening technique of subtractive smoothing applied to bands 4, 5 and 7 of a Landsat multispectral scanner image and the effect this has on the colour composite formed from these bands. As noted the sharpened image has clearer high frequency detail; however there is a tendency for noise to be enhanced, as might be expected.

## 5.7 Line Detection

### 5.7.1

#### Linear Line Detecting Templates

Line features such as rivers and roads in satellite images can be detected as pairs of edges if they are more than one pixel wide or alternatively, if they are a single pixel in width, they can be detected using the following line detecting templates:

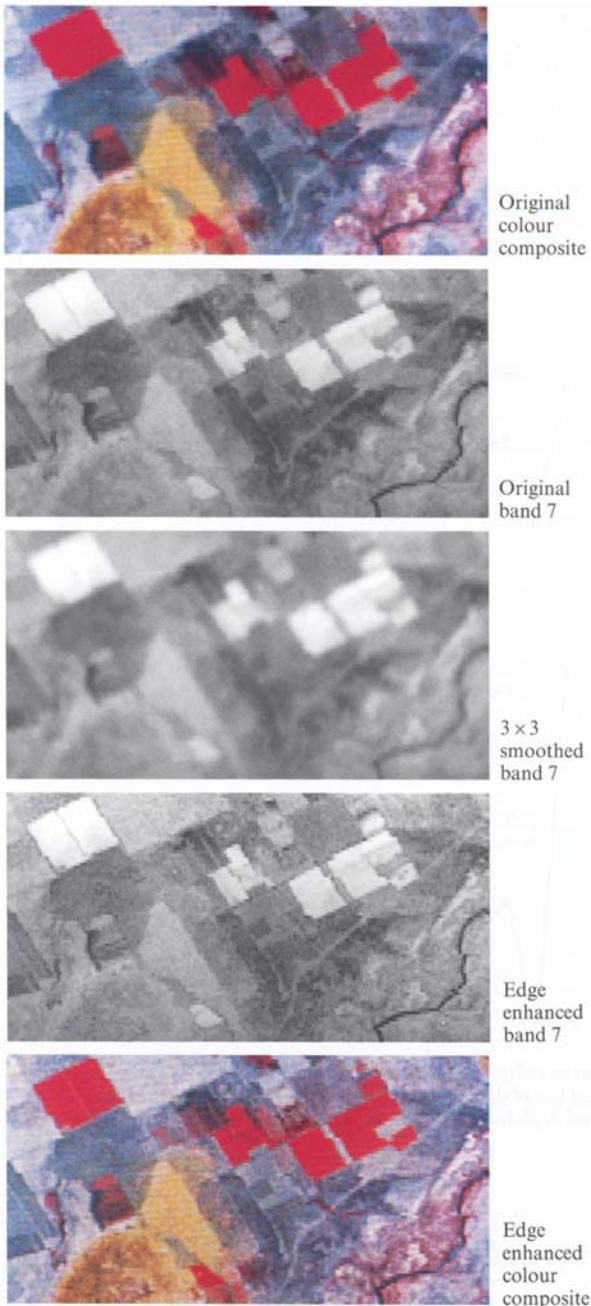
			
vertical	horizontal		diagonal

These templates seem not to have been used to any great extent in remote sensing image processing since lines, in addition to edges, are enhanced using the gradient and subtractive smoothing techniques of Sect. 5.6. Moreover, with sensor resolutions available up to 1982, not many single pixel width linear features have been apparent in imagery. With resolutions in the range of 10 m to 30 m however, cultural features such as roads, could be amenable to detection using line related templates.

### 5.7.2

#### Non-linear and Semi-linear Line Detecting Templates

The line detecting templates of Sect. 5.7.1 are regarded as linear since their convolution with image data is a linear mathematical operation. Some nonlinear line detecting template operations have also been proposed. To describe these it is of value to denote a  $3 \times 3$  neighbourhood of pixels in an image as



**Fig. 5.11.** Illustration of subtractive smoothing as an image sharpening procedure

$$\begin{array}{ccc} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ A_3 & B_3 & C_3 \end{array}$$

A nonlinear line detector algorithm, proposed by Rosenfeld and Thurston (1971), establishes pixel  $B_2$  as part of a dark vertical line if

$$A_i, C_i > B_i$$

by a prespecified threshold. Similar expressions apply for lines of other orientations and for bright lines on dark backgrounds.

Vanderbrug (1976) has proposed what he calls a semilinear detector. For the pixel array above this determines  $B_2$  as part of a dark vertical line if

$$\sum_{i=1}^3 A_i \quad \text{and} \quad \sum_{i=1}^3 C_i > \sum_{i=1}^3 B_i$$

by some prespecified threshold.

Gurney (1980) has noted that the semilinear detector works better than the nonlinear algorithm although line thickening results and computational cost is high. These disadvantages are obviated by the use of the additional constraint with the semilinear algorithm:

$$A_2 > B_2 \quad \text{and} \quad C_2 > B_2$$

Gurney also discusses means by which the thresholds for the semilinear detector can be effectively established.

## 5.8 General Convolution Filtering

It is clear that smoothing, edge and line detection represent just particular ways of defining the template entries in (5.1) and that more general spatial filtering operations could be defined by loading the template in different fashions. For example, edge enhancement by subtractive smoothing treated in Sect. 5.6.4 could be implemented by the single template

$$\begin{array}{c} \begin{array}{|c|c|c|} \hline -a & -a & -a \\ \hline -a & 2-a & -a \\ \hline -a & -a & -a \\ \hline \end{array} \end{array} = \begin{array}{c} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \end{array} + \begin{array}{c} \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} \end{array} - \begin{array}{c} \begin{array}{|c|c|c|} \hline a & a & a \\ \hline a & a & a \\ \hline a & a & a \\ \hline \end{array} \end{array}$$

where  $a=1/9$ . This template implements a high spatial frequency boosting.

By expanding the size of the template it is possible to determine detectors that are sensitive to edges and lines in other than the four common orientations. In addition, templates can be used for recognition of large objects in imagery, where the templates

are loaded with zeros, except for those locations corresponding to the shape and orientation of an object of interest. In this case the procedure is referred to as template matching and is more akin to correlation than convolution (Rosenfeld, 1978).

## 5.9

### Detecting Geometric Properties

A number of procedures can be devised that allow geometric properties in images to be detected and measured. While they are not geometric enhancement operations as such, they share the common theme with the methods treated previously in this chapter in that they require neighbourhood operations for their computation.

#### 5.9.1

##### Texture

We all know what texture is – we can clearly see the different textures present in images, but quantitative characterisation of texture is not simple. First, it is necessary to find a measure that somehow captures the spatial properties of a scene that reveal texture. A long-standing measure is the grey level co-occurrence matrix (GLCM) defined in the following way (Haralick, 1979). To make the development simple, imagine we want to detect a component of texture just in the horizontal direction in a particular region of an image. To do this we could see how often two particular grey levels in the image occur in that direction in the selected region, separated by a given distance. We could then look for the same sort of behaviour in other directions, such as vertically and diagonally, in which case there would be four matrices for any chosen pixel separation. This suggests that what we are looking for can be characterised by some form of repeating pattern which, of course, is what texture is.

Let  $g(\phi_1, \phi_2 | h, \theta)$  be the relative occurrence of pixels with grey levels  $\phi_1$  and  $\phi_2$  spaced  $h$  pixels apart, in the direction  $\theta$  – here chosen as horizontal. Relative occurrence is the number of times each grey level pair is counted divided by the total possible number of grey level pairs. The GLCM for a region, defined by a user-specified window, is the matrix of those measurements over all grey level pairs. If there are  $L$  brightness values possible then the GLCM will be an  $L \times L$  matrix. Note there will be one GLCM for each of the chosen values of  $h$  and  $\theta$ . Given that  $L$  can be quite large for some sensors ( $L = 1024$  for 10 bit data) sometimes the brightness value range is either restricted or its dynamic range is reduced by considering the co-occurrence of brightness value in ranges.

There will be as many GLCMs as there are values chosen for  $h$  and  $\theta$ . Often  $h$  is used as a variable to see whether texture exists on a local or more regional scale in an image. On the other hand the GLCMs computed for various values of  $\theta$  are either kept separate to see whether the texture is orientation dependent, or they are averaged on the assumption that texture will not vary significantly with orientation.

Once we have the GLCMs for the regions of interest it is then appropriate to set up a single metric computed from each matrix to use as a texture measure. A range

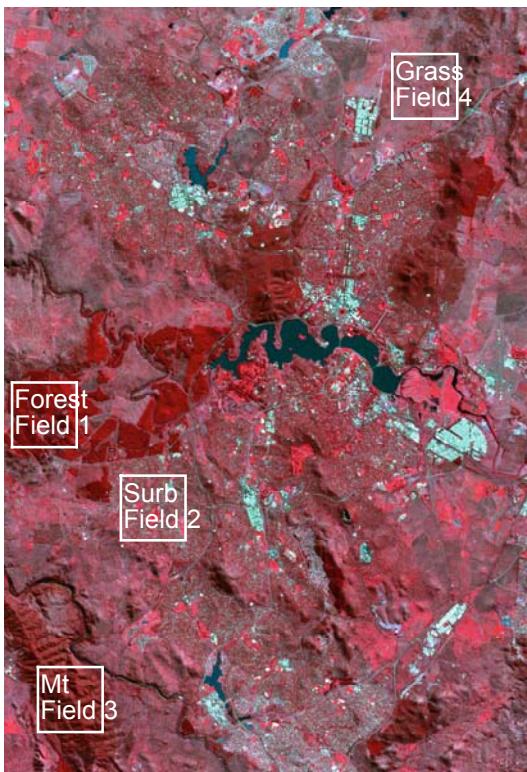
of measures is possible one of which is to describe the *entropy* of the information contained in the GLCM, defined by

$$H = - \sum_{\phi_1=1}^L \sum_{\phi_2=1}^L g(\phi_1, \phi_2 | h, \theta) \log [g(\phi_1, \phi_2 | h, \theta)]$$

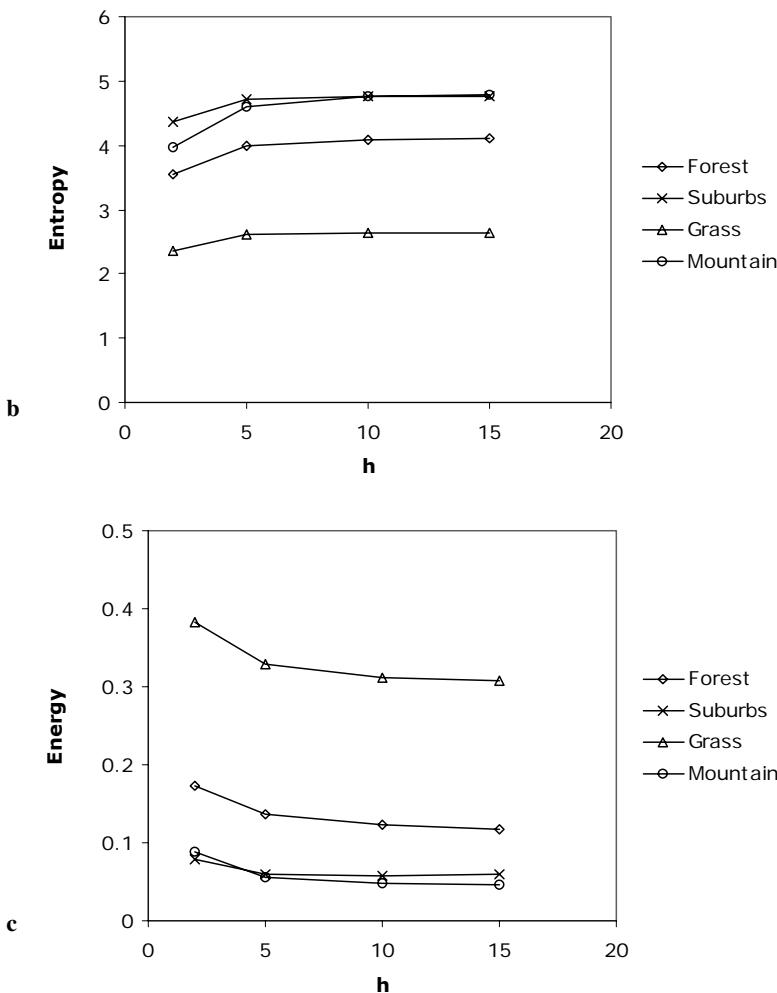
Entropy will be highest when all entries in the GLCM are equi-probable, ie when the image is not obviously textured, and will be low when there is a large disparity in the probabilities, as might be expected when significant texture is present.

Another measure is *energy* which is the sum of the squared elements of the GLCM. It will be small when the GLCM elements are small, indicating low texture.

Figure 5.12a shows an ETM+ image of a region surrounding Canberra, the Federal Capital of Australia. Four small regions are indicated as “fields” by white rectangles, within which just the horizontal GLCMs were computed for a range of values of lag,  $h$ . Those calculations used just the first ETM+ band – i.e. the visible blue, which was reduced in dynamic range to 32 bits before any calculations were performed.



**Fig. 5.12a.** Portion of an ETM+ image in the region surrounding Canberra, showing four fields used as regions for the computation of grey level co-occurrence matrices and subsequent texture properties



**Fig. 5.12. b** Entropy as a function of pixel separation (or lag); **c** Energy as a function of pixel separation

Figures 5.12b and c show the variation of entropy and energy with lag. Two points are noteworthy. First, entropy increases with lag and energy decreases with lag, indicating that the texture is falling away at larger spacings. Secondly the four cover types chosen – grass, forest, mountain and suburban are separable by their texture, with grassland exhibiting the strongest texture. The suburban and mountainous regions are seen to be low in texture by comparison, and are comparable to each other for the range of scales chosen. Note that entropy and energy behave oppositely to each other as might be expected.

### 5.9.2

#### Spatial Correlation – The Semivariogram

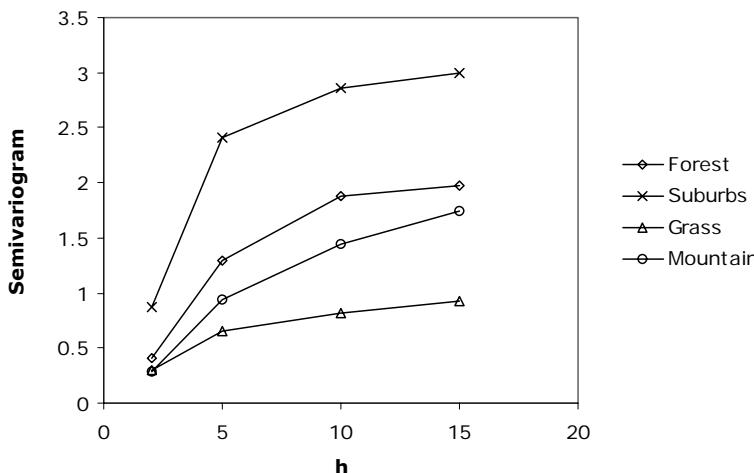
The semivariogram is a useful means for describing the spatial properties of an image (or the scene being imaged) in a specified direction. It is constructed in the following manner, by computing the average semivariance in the given direction according to (Curran, 1988)

$$\overline{S^2} = \frac{1}{2m} \sum_{i=1}^m [\phi(i) - \phi(i + h)]^2$$

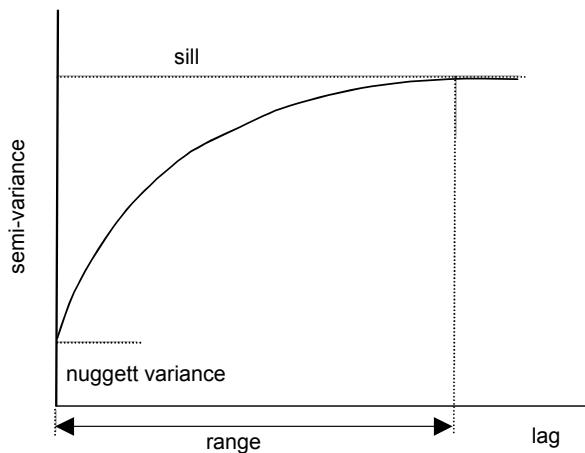
In this expression  $h$  represents the distance, or lag, between two pixels whose brightness values,  $\phi(i)$  and  $\phi(i + h)$  are subtracted and then squared. By moving along the given direction by pixel ( $i = 1, 2, \dots, m$ ) one half of the averaged squared distance is computed, as indicated in the formula;  $m$  is the total number of pairs which are separated by  $h$ . By varying the lag,  $h$ , a graph of the average semivariance versus lag can be constructed as depicted in Fig. 5.13. That graph is called a semivariogram. In a sense the semivariance measure is detecting how dissimilar pixel brightnesses are, on the average, when separated by a lag  $h$ . If there is spatial periodicity in the landscape then the semivariogram will reflect that behaviour as well.

The semivariogram for the image of Fig. 5.12a is shown in Fig. 5.13 for the four regions chosen.

Several properties can be derived from the semivariogram; these are best illustrated by the idealised form shown in Fig. 5.14, and include the *sill* (its asymptotic maximum value, if it exists), the *nugget variance* (the extrapolated point of intersection with the ordinate), sometimes taken to indicate the noise properties of the image since it represents variance that is not related to the spatial properties of the scene, and the *range*, which is the lag or separation at which the sill is reached.



**Fig. 5.13.** Semivariogram for the image of Fig. 5.12a



**Fig. 5.14.** The idealised semivariogram for a region in which there is no spatial periodicity

### 5.9.3

#### Shape Detection

Recognition of shapes in image data has not been considered in remote sensing as extensively as it has in object recognition exercises, such as robot vision. Presumably this is because the resolution generally available in the past has been insufficient to define shape with any degree of precision. However with ground resolution elements better than 20 to 30 m in imagery, shapes such as those of rectangular fields in agriculture, and circular pivotal irrigation systems are quite well defined.

Shape recognition can be carried out using template techniques, in which the templates are chosen according to the shape of interest (Hord, 1982). The operation required is one of correlation and not the convolution operation of (5.4). Correlation is defined by that same expression but with additions in place of subtractions. A major difficulty with this approach, which as a consequence renders the technique of limited value in practice, is that the template must match not only the shape of interest, but also its size and orientation. Other methods therefore are often employed. These include the adoption of shape factors (Underwood, 1970), moments of area (Pavlidis, 1978) and Fourier transforms of shape boundaries (Pavlidis, 1980). In each of these the shape must first be delineated from the rest of the image. This is achieved by edge and line detection processes.

## References for Chapter 5

The template techniques that form the basis of much of the material presented in this chapter are treated also by Castleman (1996), Moik (1980) and Hord (1982). Castleman also provides a detailed linear systems theory approach to filter design. Gonzalez and Woods (1992) present the method in a vector formulation, noting that the convolution operation in (5.1) can be expressed

as the scalar product of the vector of template entries and the vector of pixel brightnesses currently covered by the template. Such an approach allows templates to be designed to detect combinations of lines and edges. Since the template entries are expressed in vector form they can be used to define vector sub-spaces into which an image has projections. A large projection into an edge sub-space implies edges in the image of the pixel currently being assessed, and so on. This is assessed in terms of the vector angle between the image pixel vector and the subspace basis vectors (template entries).

Gradient methods are covered also by Moik (1980), Gonzalez and Woods (1992), Hord (1982) and to an extent by Castleman (1996). Gonzalez and Woods also include discussions on the use of thresholds applied to the response of gradient operators. Vanderbrug (1976) and Gurney (1980) consider the properties of nonlinear and semilinear line detecting templates.

Paine and Lodwick (1989) provide a good discussion of the application of edge detection methods, while Brzakovic et al. (1991) consider the use of rule-based methods to assist in edge detection when a number of templates is involved. While the edge detection methods treated here have been applicable only to single bands of data, Cumani (1991) and Drewnick (1994) have proposed operators for use on multispectral data.

- K.R. Babu and R. Nevatia, 1980: Linear Feature Extraction and Description. *Computer Graphics and Image Processing*, 13, 257–269.
- E.O. Brigham, 1974: *The Fast Fourier Transform*, N.J. Prentice-Hall.
- E.O. Brigham, 1988: *The Fast Fourier Transform and its Applications*, N.J. Prentice-Hall.
- D. Brzakovic, R. Patton and R.L. Wang, 1991: Rule-based Multitemplate Edge Detector. *CVGIP: Graphical Models and Image Processing*, 53, 258–268.
- K.R. Castleman, 1996: *Digital Image Processing*, N.J. Prentice-Hall.
- A. Cumani, 1991: Edge Detection in Multispectral Images. *CVGIP: Computer Models and Image Processing*, 53, 40–51.
- C. Drewnick, 1994: Multispectral Edge Detection. Some Experiments on Data from Landsat TM. *Int. J. Remote Sensing*, 15, 3743–3765.
- E.M. Eliason and A.S. McEwan, 1990: Adaptive Box Filters for Removal of Random Noise from Digital Images. *Photogrammetric Engineering and Remote Sensing*, 56, 453–458.
- R.C. Gonzalez and R.E. Woods, 1992: *Digital Image Processing*, Mass., Addison-Wesley.
- C.M. Gurney, 1980: Threshold Selection for Line Detection Algorithms. *IEEE Trans. Geoscience and Remote Sensing*, GE-18, 204–211.
- R.M. Haralick, 1979: Statistical and Structural Approaches to Texture. *Proc. IEEE*, 67, 786–802.
- R.M. Hord, 1982: *Digital Image Processing of Remotely Sensed Data*, N.Y. Academic.
- C.D. McGillem and G.R. Cooper, 1984: *Continuous and Discrete Signal and Systems Analysis*, 2e, N.Y., Holt, Reinhard and Winston.
- J.G. Moik, 1980: *Digital Processing of Remotely Sensed Images*, N.Y., Academic.
- S.H. Paine and G.D. Lodwick, 1989: Edge Detection and Processing of Remotely Sensed Digital Images. *Photogrammetria (PRS)*, 43, 323–336.
- C. Paul and K.S. Shanmugan, 1982: A Fast Thinning Operator. *IEEE Trans. Systems, Man, Cybernetics*, SMC-12, 567–569.
- T. Pavlidis, 1978: A Review of Algorithms for Shape Analysis. *Computer Graphics and Image Processing*, 7, 243–258.
- T. Pavlidis, 1980: Algorithms for Shape Analysis of Contours and Waveforms. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-2, 301–312.
- A. Rosenfeld, 1978: *Image Processing and Recognition*, Technical Report 664, Computer Vision Laboratory, University of Maryland.

A. Rosenfeld and M. Thurston, 1971: Edge and Curve Detection for Visual Scene Analysis, IEEE Trans. Computers, C-20, 562–569.

E.E. Underwood, 1970: Quantitative Stereology, Mass., Addison-Wesley.

G.J. Vanderbrug, 1976: Line Detection in Satellite Imagery, IEEE Trans. Geoscience Electronics, GE-14, 37–44.

## Problems

**5.1** The template entries for line and edge detection sum to zero whereas those for smoothing do not. Why do you think that is so?

**5.2** Repeat the example of Fig. 5.10 but by using a  $[5 \times 1]$  smoothing operation in part (a), rather than  $[3 \times 1]$  smoothing.

**5.3** Repeat the example of Fig. 5.10 but by using a  $[3 \times 1]$  median filtering operation in part (a) rather than  $[3 \times 1]$  mean value smoothing.

**5.4** An alternative smoothing process to median and mean value filtering using template methods is known as modal filtering. In this approach a pixel at the centre of the template neighbourhood is replaced by the brightness value that occurs most frequently in the neighbourhood. Apply  $[3 \times 1]$  and  $[5 \times 1]$  modal filters to the image data of Fig. 5.6. Note differences in the results compared with mean value and median smoothing, particularly around the edges.

**5.5** Suppose  $S$  is a template operation that implements smoothing and  $O$  is the template operator that leaves an image unchanged (see Sect. 5.8). Then an edge enhanced image created by the subtractive smoothing approach of Sect. 5.6.4 can be expressed according to

$$\text{New image} = O(\text{old image}) + O(\text{old image}) - S(\text{old image})$$

Rewrite this expression to incorporate two user defined parameters  $\alpha$  and  $\beta$  that will cause the formula to implement any of smoothing, edge detection or edge enhancement.

**5.6** (Requires vector algebra background). Show that template methods for line and edge detection can be expressed as the scalar product of a vector composed from the template entries and a vector formed from the neighbourhood of pixels currently covered by the template. Show how the angle between the template and pixel vectors can be used to assess the edge or line feature a current pixel most closely corresponds to. (See Gonzalez and Woods (1992)).

**5.7** The following kernel is sometimes convolved with image data. What operation will it implement?

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

**5.8** Consider the middle pixel shown in the figure below and calculate its new value if

- (i) a  $3 \times 3$  median filtering is applied,
- (ii) a  $3 \times 3$  template which performs edge enhancement by subtractive smoothing is applied,

- (iii) a  $3 \times 1$  image smoothing template with a threshold 2 is applied,
- (iv) the Sobel operator is applied for edge detection.

.	.	.	.	.
.	3	7	0	.
.	8	1	1	.
.	7	2	9	.
.	.	.	.	.

**5.9** Image smoothing can be performed by template operators that implement averaging or median filtering. Compare the methods, particularly as they affect edges. Would you expect median filtering to be useful in edge enhancement by the technique of subtracting a smoothed image from the original?

**5.10** If a  $3 \times 3$  smoothing template is applied to an image twice in succession, how many neighbours will have played a part in modifying the brightness of a given pixel? Design a single template to achieve the same result in one pass.

# 6

## Multispectral Transformations of Image Data

The multispectral or vector character of most remote sensing image data renders it amenable to spectral transformations that generate new sets of image components or bands. These components then represent an alternative description of the data, in which the new components of a pixel vector are related to its old brightness values in the original set of spectral bands via a linear operation. The transformed image may make evident features not discernable in the original data or alternatively it might be possible to preserve the essential information content of the image (for a given application) with a reduced number of the transformed dimensions. The last point has significance for displaying data in the three dimensions available on a colour monitor or in colour hardcopy, and for transmission and storage of data.

The role of this chapter is to present image transformations of value in the enhancement of remote sensing imagery, although some also find application in pre-conditioning image data prior to classification by the techniques of Chaps. 8 and 9. The techniques covered, which appeal directly to the vector nature of the image, include the principal components transformation and so-called band arithmetic. The latter includes the creation of ratio images. Some specialised transformations, such as the Kauth-Thomas tasseled cap transform are also treated.

### 6.1

#### The Principal Components Transformation

The multispectral or multidimensional nature of remote sensing image data can be accommodated by constructing a vector space with as many axes or dimensions as there are spectral components associated with each pixel. In the case of Landsat Thematic Mapper data it will have seven dimensions while for SPOT HRV data it will be three dimensional. For hyperspectral data there may be several hundred axes. A particular pixel in an image is plotted as a point in such a space with co-ordinates that correspond to the brightness values of the pixels in the appropriate spectral components. For simplicity the treatment to be developed in this topic will

be based upon a two dimensional multispectral space (say visible red and infrared) since the diagrams are then easily understood and the mathematical detail is readily assimilated. The results derived however are perfectly general and apply to data of any dimensionality.

### 6.1.1

#### The Mean Vector and Covariance Matrix

The positions of pixel points in multispectral space can be described by vectors, whose components are the individual spectral responses in each band. Strictly, these are vectors drawn from the origin to the pixel point as seen in Appendix D, but this concept is not used explicitly. Consider a multispectral space with a large number of pixels plotted in it as shown in Fig. 6.1, with each pixel described by its appropriate vector  $\mathbf{x}$ . The mean position of the pixels in the space is defined by the expected value of the pixel vector  $\mathbf{x}$ , according to

$$\mathbf{m} = \mathbb{E}\{\mathbf{x}\} = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k \quad (6.1)$$

where  $\mathbf{m}$  is the mean pixel vector and the  $\mathbf{x}_k$  are the individual pixel vectors of total number  $K$ ;  $\mathbb{E}$  is the expectation operator.

While the mean vector is useful to define the average or expected position of the pixels in multispectral vector space, it is of value to have available a means by which their scatter or spread is described. This is the role of the covariance matrix which is defined as

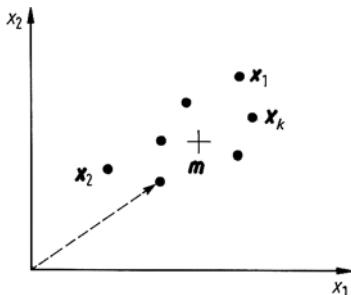
$$\Sigma_x = \mathbb{E}\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^t\} \quad (6.2a)$$

in which the superscript ‘ $t$ ’ denotes vector transpose. (See Appendix D).

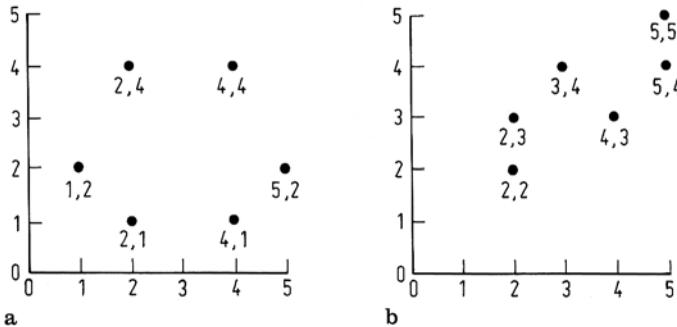
An unbiased estimate of the covariance matrix is given by

$$\Sigma_x = \frac{1}{K-1} \sum_{k=1}^K (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^t \quad (6.2b)$$

The covariance matrix is one of the most important mathematical concepts in the analysis of multispectral remote sensing data, as a result of which it is of value



**Fig. 6.1.** Two dimensional multispectral space showing the individual pixel vectors and their mean position, as defined by  $\mathbf{m}$ , the mean vector



**Fig. 6.2.** Two dimensional data showing no correlation between components **a** and high correlation between components **b**

to consider some sample calculations to enable its properties to be emphasised. In these it will be seen that if there is correlation between the responses in a pair of spectral bands the corresponding off-diagonal element in the covariance matrix will be large by comparison to the diagonal terms. On the other hand, if there is a little correlation, the off-diagonal terms will be close to zero. This behaviour can also be described in terms of the correlation matrix  $R$  whose elements are related to those of the covariance matrix by

$$\rho_{ij} = v_{ij} / \sqrt{v_{ii} v_{jj}} \quad (6.3)$$

where  $\rho_{ij}$  is an element of the correlation matrix and  $v_{ij}$  etc. are elements of the covariance matrix;  $v_{ii}$  and  $v_{jj}$  are the variances of the  $i$ th and  $j$ th bands of data. The  $\rho_{ij}$  describe the correlation between band  $i$  and band  $j$ .

Consider the two, two-dimensional sets of data shown in Fig. 6.2. That in Fig. 6.2a shows little correlation between the two components: in other words, both components are necessary to describe where a pixel lies in the space. The data shown in Fig. 6.2b however exhibits a high degree of correlation between its two components, evident in the elongated spread of the data at an angle to the axes. One dimension on its own is almost sufficient to predict where a pixel lies in the space, and an increase or decrease in either component suggests a corresponding increase or decrease in the other. This is not the case with Fig. 6.2a. In terms of the individual images corresponding to the bands of multispectral data, highly correlated bands as depicted in Fig. 6.2b would yield image components very similar in appearance. Where one is dark the other will be dark and so on. The image components corresponding to Fig. 6.2a, however, would display no similar consistently common behaviour. Problem 6.7 shows a number of other situations of high and low correlation. Importantly, if the data is scattered in an elongated fashion, as seen in Fig. 6.2b, but the directions of major scatter are parallel to the coordinate (measurement) axes, then there is little correlation among the measurements.

Table 6.1 shows a sample set of hand calculations undertaken to find the covariance and correlation matrices for Fig. 6.2a. Normally this would be carried out by computer, particularly for data with higher dimensionality. As noted from the corre-

**Table 6.1.** Computation of covariance and correlation matrices for Fig. 6.2a

The mean vector is  $\mathbf{m} = \begin{bmatrix} 3.00 \\ 2.33 \end{bmatrix}$

$x$	$x - \mathbf{m}$	$[x - \mathbf{m}] [x - \mathbf{m}]^T$
$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$	$\begin{bmatrix} -2.00 \\ -0.33 \end{bmatrix}$	$\begin{bmatrix} 4.00 & 0.66 \\ 0.66 & 0.11 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} -1.00 \\ -1.33 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 1.33 \\ 1.33 & 1.77 \end{bmatrix}$
$\begin{bmatrix} 4 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 1.00 \\ -1.33 \end{bmatrix}$	$\begin{bmatrix} 1.00 & -1.33 \\ -1.33 & 1.77 \end{bmatrix}$
$\begin{bmatrix} 5 \\ 2 \end{bmatrix}$	$\begin{bmatrix} 2.00 \\ -0.33 \end{bmatrix}$	$\begin{bmatrix} 4.00 & -0.66 \\ -0.66 & 0.11 \end{bmatrix}$
$\begin{bmatrix} 4 \\ 4 \end{bmatrix}$	$\begin{bmatrix} 1.00 \\ 1.67 \end{bmatrix}$	$\begin{bmatrix} 1.00 & 1.67 \\ 1.67 & 2.79 \end{bmatrix}$
$\begin{bmatrix} 2 \\ 4 \end{bmatrix}$	$\begin{bmatrix} -1.00 \\ 1.67 \end{bmatrix}$	$\begin{bmatrix} 1.00 & -1.67 \\ -1.67 & 2.79 \end{bmatrix}$
whereupon		$\Sigma_x = \begin{bmatrix} 2.40 & 0 \\ 0 & 1.87 \end{bmatrix}$
and		$R = \begin{bmatrix} 1.00 & 0 \\ 0 & 1.00 \end{bmatrix}$
where $R$ is the correlation matrix.		

lation matrix there is no correlation between the individual components of the data, a fact which is evident also in the zero off-diagonal entries in the covariance matrix. The entry of 2.40 in the upper left hand corner of the covariance matrix signifies that the data points have a variance of 2.40 along the horizontal axis, or a standard deviation of 1.55 about the mean. Similarly, the variance and standard deviation vertically are 1.87 and 1.37 respectively.

For the data in Fig. 6.2b, it is shown by a similar set of calculations to those in Table 6.1 that

$$\mathbf{m} = \begin{bmatrix} 3.50 \\ 3.50 \end{bmatrix} \quad \Sigma_x = \begin{bmatrix} 1.900 & 1.100 \\ 1.100 & 1.100 \end{bmatrix}$$

and

$$R = \begin{bmatrix} 1.000 & 0.761 \\ 0.761 & 1.000 \end{bmatrix}$$

Thus components 1 and 2 of the data in Fig. 6.2b are 76% correlated.

It should be noted that both the covariance and correlation matrices are symmetric and that an image data set, in which there is no correlation between any of its multispectral components, will have a diagonal covariance (and correlation) matrix.

### 6.1.2

#### A Zero Correlation, Rotational Transform

It is fundamental to the development of the principal components transformation to ask whether there is a new co-ordinate system in the multispectral vector space in which the data can be represented without correlation; in other words, such that the covariance matrix in the new co-ordinate system is diagonal. For a particular two dimensional vector space such a new co-ordinate system is depicted in Fig. 6.3. If the vectors describing the pixel points are represented as  $\mathbf{y}$  in the new co-ordinate system then it is desired to find a linear transformation  $G$  of the original co-ordinates, such that

$$\mathbf{y} = G\mathbf{x} = D^t \mathbf{x} \quad (6.4)$$

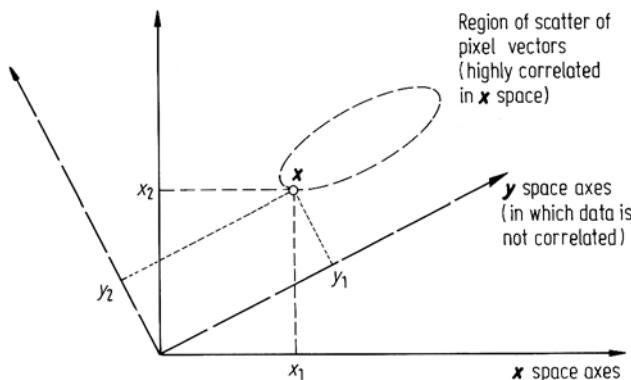
subject to the constraint that the covariance matrix of the pixel data in  $\mathbf{y}$  space is diagonal. Expressing  $G$  as  $D^t$  will make the comparison of principal components with other transformation operations, treated later, much simpler.

In  $\mathbf{y}$  space the covariance matrix is, by definition,

$$\Sigma_y = \mathbb{E}\{(\mathbf{y} - \mathbf{m}_y)(\mathbf{y} - \mathbf{m}_y)^T\}$$

where  $\mathbf{m}_y$  is the mean vector expressed in terms of the  $\mathbf{y}$  co-ordinates. It is shown readily that

$$\mathbf{m}_y = \mathbb{E}\{\mathbf{y}\} = \mathbb{E}\{D^t \mathbf{x}\} = D^t \mathbb{E}\{\mathbf{x}\} = D^t \mathbf{m}_x {}^1$$



**Fig. 6.3.** Illustration of a modified co-ordinate system in which the pixel vectors have uncorrelated components

---

<sup>1</sup>  $\mathbb{E}\{D^t \mathbf{x}\} = \frac{1}{K} \sum_{k=1}^K D^t \mathbf{x}_k = D^t \frac{1}{K} \sum_{k=1}^K \mathbf{x}_k = D^t \mathbf{m}_x$

i.e.  $D^t$ , being a matrix of constants, can be taken outside an expectation operator.

where  $\mathbf{m}_x$  is the data mean in  $\mathbf{x}$  space. Therefore

$$\Sigma_y = \mathcal{E}\{(D^t \mathbf{x} - D^t \mathbf{m}_x)(D^t \mathbf{x} - D^t \mathbf{m}_x)^t\}$$

which can be written as

$$\Sigma_y = D^t \mathcal{E}\{(\mathbf{x} - \mathbf{m}_x)(\mathbf{x} - \mathbf{m}_x)^t\} D^2$$

$$\text{i.e. } \Sigma_y = D^t \Sigma_x D \quad (6.5)$$

where  $\Sigma_x$  is the covariance of the pixel data in  $\mathbf{x}$  space. Since  $\Sigma_y$  must, by demand, be diagonal,  $D$  can be recognised as the matrix of eigenvectors of  $\Sigma_x$ , provided  $D$  is an orthogonal matrix. This can be seen from the material presented in Appendix D dealing with the diagonalization of a matrix. As a result,  $\Sigma_y$  can then be identified as the diagonal matrix of eigenvalues of  $\Sigma_x$ ,

$$\Sigma_y = \begin{bmatrix} \lambda_1 & 0 & & \\ 0 & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_N \end{bmatrix}$$

where  $N$  is the dimensionality of the data. Since  $\Sigma_y$  is, by definition, a covariance matrix and is diagonal, its elements will be the variances of the pixel data in the respective transformed co-ordinates. It is arranged such that  $\lambda_1 > \lambda_2 > \dots > \lambda_N$  so that the data exhibits maximum variance in  $y_1$ , the next largest variance in  $y_2$  and so on, with minimum variance in  $y_N$ .

The principal components transform defined by (6.4) subject to the diagonal constraint of (6.5) is also known as the Karhunen-Loëve or Hotelling transform.

Before proceeding it is of value at this stage to pursue further the examples of Fig. 6.2, to demonstrate the computational aspects of principal components analysis. Recall that the original  $\mathbf{x}$  space covariance matrix for the highly correlated image data of Fig. 6.2b is

$$\Sigma_x = \begin{bmatrix} 1.90 & 1.10 \\ 1.10 & 1.10 \end{bmatrix}$$

To determine the principal components transformation it is necessary to find the eigenvalues and eigenvectors of this matrix. The eigenvalues are given by the solution to the characteristic equation

$$|\Sigma_x - \lambda I| = 0, \quad I \text{ being the identity matrix.}$$

$$\text{i.e. } \begin{vmatrix} 1.90 - \lambda & 1.10 \\ 1.10 & 1.10 - \lambda \end{vmatrix} = 0$$

$$\text{or } \lambda^2 - 3.0\lambda + 0.88 = 0$$

which yields  $\lambda = 2.67$  and  $0.33$

---

<sup>2</sup> Since  $[A\zeta]^t = \zeta^t A^t$  (reversed law of matrices). Note also  $[A\zeta]^{-1} = \zeta^{-1} A^{-1}$ .

As a check on the analysis it may be noted that the sum of the eigenvalues is equal to the trace of the covariance matrix, which is the sum of its diagonal elements.

The covariance matrix in the appropriate  $y$  co-ordinate system (with principal components as axes) is therefore

$$\Sigma_y = \begin{bmatrix} 2.67 & 0 \\ 0 & 0.33 \end{bmatrix}$$

Note that the first principal component, as it is called, accounts for  $2.67/(2.67 + 0.33) \equiv 89\%$  of the total variance of the data in this particular example. It is now of interest to find the actual principal components transformation matrix  $G = D^t$ . Note that this is the *transposed* matrix of eigenvectors of  $\Sigma_x$ . Consider first, the eigenvector corresponding to  $\lambda_1 = 2.67$ . This is the vector solution to the equation

$$[\Sigma_x - \lambda_1 I] g_1 = 0$$

with  $g_1 = \begin{bmatrix} g_{11} \\ g_{21} \end{bmatrix} \equiv d_1^t$  for the two dimensional example at hand.

Substituting for  $\Sigma_x$  and  $\lambda_1$  gives the pair of equations

$$-0.77g_{11} + 1.10g_{21} = 0$$

$$1.10g_{11} - 1.57g_{21} = 0$$

which are not independent, since the set is homogeneous. It does have a non-trivial solution however because the coefficient matrix has a zero determinant. From either equation it can be seen that

$$g_{11} = 1.43g_{21} \quad (6.6)$$

At this stage either  $g_{11}$  or  $g_{21}$  would normally be chosen arbitrarily, and then a value would be computed for the other. However the resulting matrix  $G$  has to be orthogonal so that  $G^{-1} \equiv G^t$ . This requires the eigenvectors to be normalised, so that

$$g_{11}^2 + g_{21}^2 = 1 \quad (6.7)$$

This is a second equation that can be solved simultaneously with (6.6) to give

$$g_1 = \begin{bmatrix} 0.82 \\ 0.57 \end{bmatrix}$$

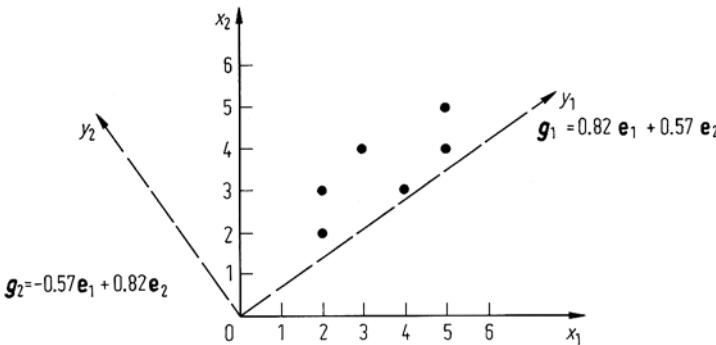
In a similar manner it can be shown that the eigenvector corresponding to  $\lambda_2 = 0.33$  is

$$g_2 = \begin{bmatrix} -0.57 \\ 0.82 \end{bmatrix}$$

The required principal components transformation matrix therefore is

$$G = D^t = \begin{bmatrix} 0.82 & -0.57 \\ 0.57 & 0.82 \end{bmatrix}^t = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix}$$

Now consider how these results can be interpreted. First of all, the individual eigenvectors  $g_1$  and  $g_2$  are vectors which define the principal component axes in



**Fig. 6.4.** Principal component axes for the data set of Fig. 6.2b;  $e_1$  and  $e_2$  are horizontal ( $x_1$ ) and vertical ( $x_2$ ) direction vectors

terms of the original co-ordinate space. These are shown in Fig. 6.4: it is evident that the data is uncorrelated in the new axes and that the new axes are a rotation of the original set. For this reason (even in more than two dimensions) the principal components transform is classed as a rotational transform.

Secondly, consider the application of the transformation matrix  $G$  to find the positions (i.e., the brightness values) of the pixels in the new uncorrelated co-ordinate system. Since  $y = Gx$  this example gives

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 0.82 & 0.57 \\ -0.57 & 0.82 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (6.8)$$

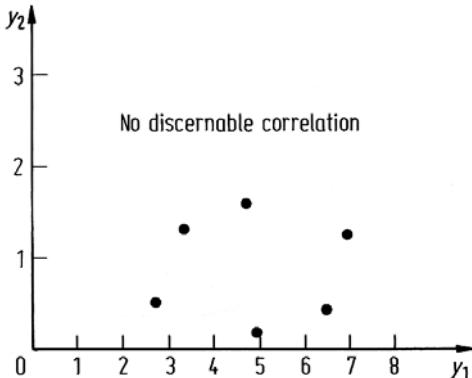
which is the actual principal components transformation to be applied to the image data. Thus, for

$$x = \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \begin{bmatrix} 5 \\ 4 \end{bmatrix}, \begin{bmatrix} 5 \\ 5 \end{bmatrix}, \begin{bmatrix} 3 \\ 4 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix}$$

we find

$$y = \begin{bmatrix} 2.78 \\ 0.50 \end{bmatrix}, \begin{bmatrix} 4.99 \\ 0.18 \end{bmatrix}, \begin{bmatrix} 6.38 \\ 0.43 \end{bmatrix}, \begin{bmatrix} 6.95 \\ 1.25 \end{bmatrix}, \begin{bmatrix} 4.74 \\ 1.57 \end{bmatrix}, \begin{bmatrix} 3.35 \\ 1.32 \end{bmatrix}.$$

The pixels plotted in  $y$  space are shown in Fig. 6.5. Several points are noteworthy. First, the data exhibits no discernable correlation between the pair of new axes (i.e., the principal components). Secondly, most of the data spread is in the direction of the first principal component. It could be interpreted that this component contains most of the information in the image. Finally, if the pair of principal component images are produced by using the  $y_1$  and  $y_2$  component brightness values for the pixels, the first principal component image will show a high degree of contrast whereas the second will have limited contrast. By comparison to the first component, the second will make use of only a few available brightness levels. It will be seen, therefore, to lack the detail of the former. While this phenomenon may not be particularly evident for a simple two dimensional example, it is especially noticeable in the fourth component of a principal component transformed Landsat multispectral scanner image as can be assessed in Fig. 6.6.



**Fig. 6.5.** Pixel points located in (uncorrelated) principal components space

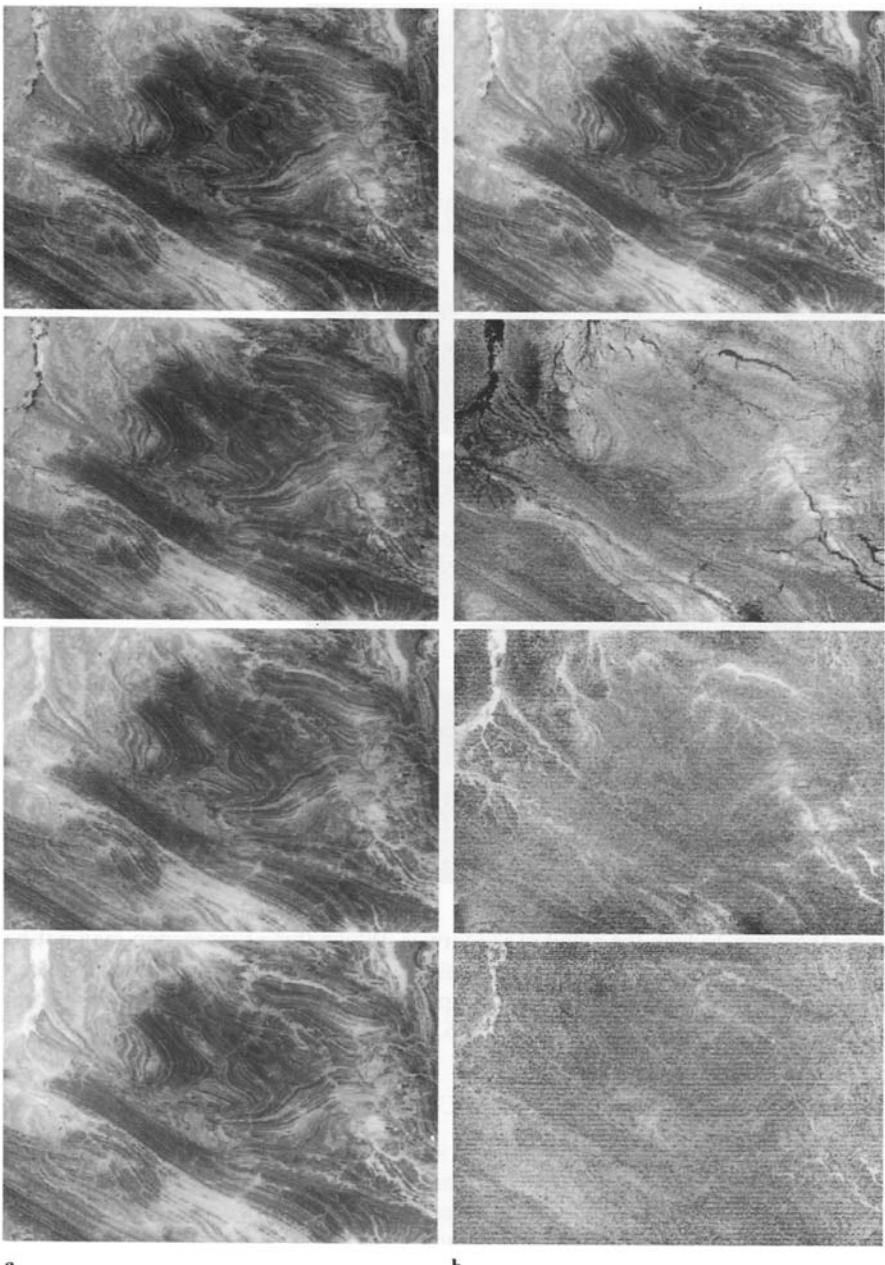
### 6.1.3

#### Examples – Some Practical Considerations

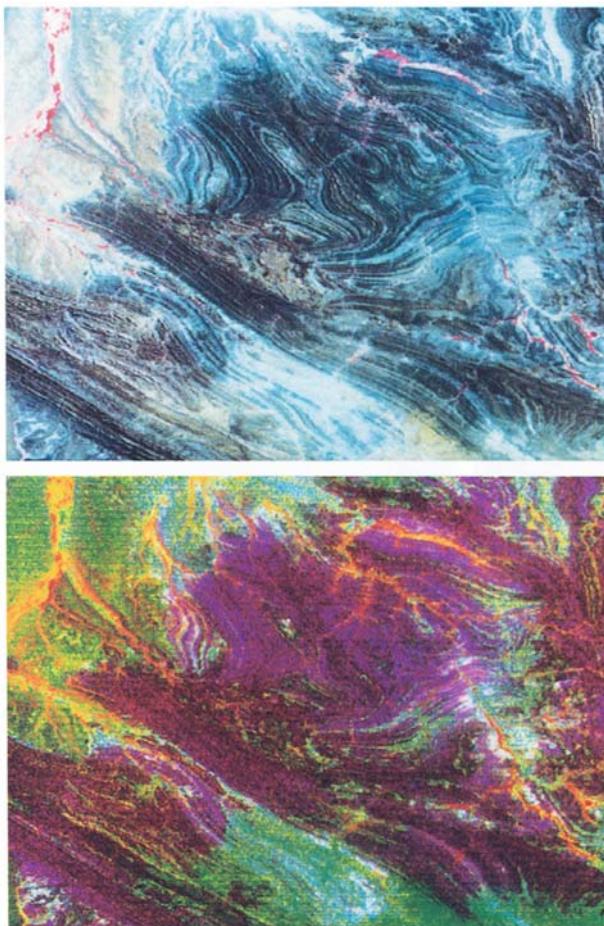
The material presented in Sect. 6.1.2 provides the background and rationale for the principal components transform. By working through the numerical example in detail the importance of eigenanalysis of the covariance matrix can be seen. However when using principal components analysis in practice the user is not involved in this level of detail. Rather only three steps are necessary, presuming software exists for implementing each of those steps. These are, first, the assembling of the covariance matrix of the image to be transformed according to (6.2). Normally, software will be available for this step, usually in conjunction with the need to generate signatures for classification as described in Chap. 8. The second step necessary is to determine the eigenvalues and eigenvectors of the covariance matrix. Either special purpose software will be available for this or general purpose matrix eigenanalysis routines can be used. The latter are found in packages such as MATLAB, Mathematica and Maple. At this stage the eigenvalues are used simply to assess the distribution of data variance over the respective components. A rapid fall off in the size of the eigenvalues indicates that the original band description of the image data exhibits a high degree of correlation and that significant results will be obtained in the transformation to follow.

The final step is to form the components using the eigenvectors of the covariance matrix as the weighting coefficients. As seen in (6.4) (noting that  $G$  is a transposed matrix of eigenvectors) and as demonstrated in (6.8), the components of the eigenvectors act as coefficients in determining the principal component brightness values for a pixel as a weighted sum of its brightnesses in the original spectral bands. The first eigenvector produces the first principal component from the original data, the second eigenvector gives rise to the second component, and so on.

Figure 6.6a shows the four original bands of an image acquired by the Landsat multispectral scanner for a small image segment in central Australia. The covariance matrix for this image is



**Fig. 6.6.** **a** Four Landsat multispectral scanner bands for the region of Andamooka in central Australia; **b** The four principal components of the image segment; **c** (overleaf) Comparison of standard false colour composite (band 7 to red, band 5 to green and band 4 to blue) with a principal component composite (first component to red, second to green and third to blue)

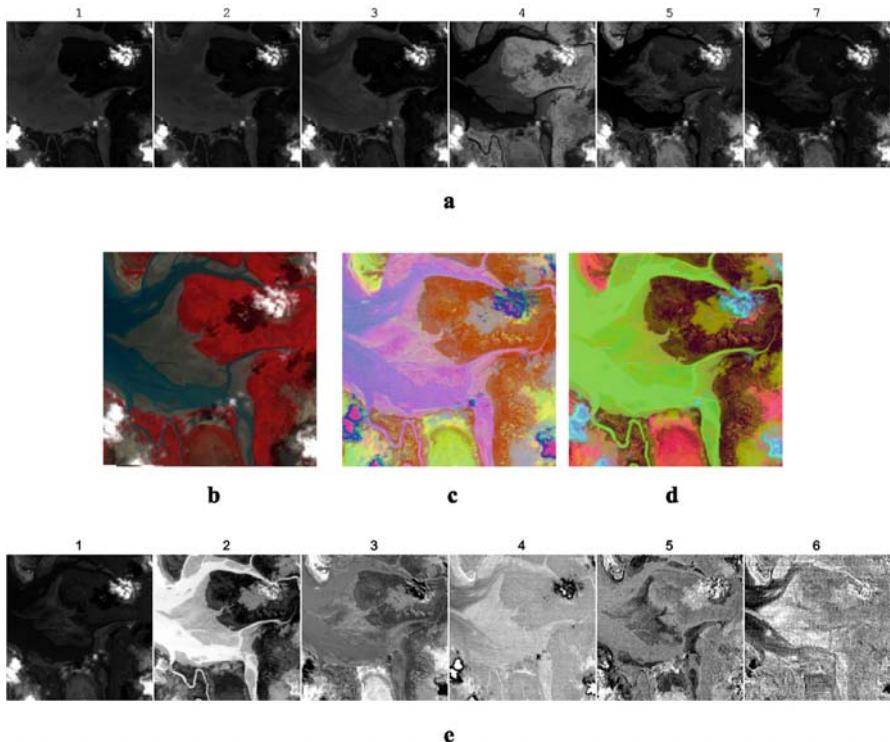
**Fig. 6.6. c**

$$\Sigma_x = \begin{bmatrix} 34.89 & 55.62 & 52.87 & 22.71 \\ 55.62 & 105.95 & 99.58 & 43.33 \\ 52.87 & 99.58 & 104.02 & 45.80 \\ 22.71 & 43.33 & 45.80 & 21.35 \end{bmatrix}$$

and its eigenvalues and eigenvectors are:

eigenvalues	253.44	7.91	3.96	0.89
eigenvector components	0.34	-0.61	0.71	-0.06
(vertically)	0.64	-0.40	-0.65	-0.06
	0.63	0.57	0.22	0.48
	0.28	0.38	0.11	-0.88

The first principal component image will be expected therefore to contain 95% of the data variance. By comparison, the variance in the last component is seen to be



**Fig. 6.7.** Principal components applied to a highly correlated TM image (without the thermal band). **a** Original TM bands; **b** Colour composite formed from TM bands 4, 3 and 2; **c** Colour composite formed from PC3, PC2 and PC1; **d** Colour composite formed from PC4, PC3 and PC2; **e** The full set of principal components.

negligible. It is to be expected that this component will appear almost totally as noise of low amplitude.

The four principal component images for this example are seen in Fig. 6.6b in which the information redistribution and compression properties of the transformation are illustrated. By association with Fig. 6.5 it would be anticipated that the later components should appear dull and poor in contrast. The high contrasts displayed are a result of a contrast enhancement applied to the components for the purpose of display. This serves to highlight the poor signal to noise ratio.

Figure 6.6c shows a comparison of a standard false colour composite formed from the original Landsat bands and a colour composite formed by displaying the first principal component as red, the second as green and the third as blue. Owing to the noise in the second and third components these were smoothed with a  $3 \times 3$  mean value template first.

A second example of the principal components transformation is shown in Fig. 6.7, this time based on the 6 reflective TM bands for a region in the Northern Territory of Australia. The covariance and correlation matrices for the image are:

$$\Sigma_x = \begin{bmatrix} 874.98 & 550.56 & 698.00 & 335.54 & 858.15 & 551.21 \\ 550.56 & 363.82 & 454.79 & 230.30 & 558.88 & 358.38 \\ 689.00 & 454.79 & 580.63 & 288.11 & 747.97 & 471.72 \\ 335.54 & 230.30 & 288.11 & 722.46 & 742.35 & 387.61 \\ 858.15 & 558.88 & 747.97 & 742.35 & 1544.70 & 871.29 \\ 551.21 & 358.38 & 471.72 & 387.61 & 871.29 & 514.18 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1.00 & 0.98 & 0.97 & 0.42 & 0.74 & 0.82 \\ 0.98 & 1.00 & 0.99 & 0.45 & 0.75 & 0.83 \\ 0.97 & 0.99 & 1.00 & 0.44 & 0.79 & 0.86 \\ 0.42 & 0.45 & 0.44 & 1.00 & 0.70 & 0.64 \\ 0.74 & 0.75 & 0.79 & 0.70 & 1.00 & 0.98 \\ 0.82 & 0.83 & 0.86 & 0.64 & 0.98 & 1.00 \end{bmatrix}$$

By computing the correlation matrix explicitly we can see how likely it is that the principal components transformation will generate new features quite different from the recorded measurement vectors. As seen, there is a high degree of correlation among the bands, so the effect of applying the principal components transformation should be quite significant. The corresponding eigenvalues and eigenvectors are:

eigenvalues	3727.35	613.34	226.14	23.52	8.16	2.25
eigenvectors	first	second	third	fourth	fifth	sixth
	0.433	0.485	-0.307	-0.684	-0.089	0.088
	0.282	0.294	-0.218	0.369	0.094	-0.801
	0.364	0.347	-0.127	0.627	-0.153	0.561
	0.303	-0.673	-0.671	0.018	0.042	0.056
	0.615	-0.322	0.562	-0.052	-0.429	-0.129
	0.362	-0.047	0.275	-0.026	0.880	0.127

Figure 6.7a shows the original TM bands, while Fig. 6.7e shows the 6 principal component images. Figure 6.7b shows a colour composite formed by mapping the original bands 4, 3, and 2 to red, green and blue respectively. Figure 6.7c shows PC3, PC2 and PC1 mapped to red, green and blue, while Fig. 6.7d shows PC4, PC3 and PC2 mapped to red, green and blue. Interestingly, the PC4, PC3, PC2 colour composite shows more detail for those ground covers whose spectral responses are dominant in the visible to near infrared regions, since PC4 (determined by the fourth eigenvector) is largely a difference image in the visible region. In contrast PC1 is essentially just a total brightness image, as can be seen from the first eigenvector, so that it does little to enhance spectral differences.

Notwithstanding the anticipated negligible information content of the last, or last few, image components resulting from a principal components analysis it is important to examine all components since often local detail may appear in a later component. The covariance matrix used to generate the principal component transformation matrix is a global measure of the variability of the original image segment. Abnormal local detail therefore may not necessarily be mapped into one of the earlier components but could just as easily appear later. This is often the case with geological structure.

### 6.1.4

#### The Effect of an Origin Shift

It will be evident that some principal component pixel brightnesses could be negative owing to the fact that the transformation is a simple axis rotation. Clearly a combination of positive and negative brightnesses cannot be displayed. Nor can negative brightness pixels be ignored since their appearance relative to the other pixels in a component serve to define detail. In practice, the problem with negative values is accommodated by shifting the origin of the principal components space to yield all components with positive and thus displayable brightnesses. This has no effect on the properties of the transformation as can be seen by inserting an origin shift term in the definition of the covariance matrix in the principal components axes. Define  $\mathbf{y}' = \mathbf{y} - \mathbf{y}_0$  where  $\mathbf{y}_0$  is the position of a new origin. In the new  $\mathbf{y}'$  co-ordinates

$$\Sigma_{\mathbf{y}'} = \mathcal{E}\{(\mathbf{y}' - \mathbf{m}_{\mathbf{y}'})(\mathbf{y}' - \mathbf{m}_{\mathbf{y}'})^t\}$$

Now  $\mathbf{m}_{\mathbf{y}'} = \mathbf{m}_{\mathbf{y}} - \mathbf{y}_0$  so that

$$\mathbf{y}' - \mathbf{m}_{\mathbf{y}'} = \mathbf{y} - \mathbf{y}_0 - \mathbf{m}_{\mathbf{y}} + \mathbf{y}_0 = \mathbf{y} - \mathbf{m}_{\mathbf{y}}.$$

Thus  $\Sigma_{\mathbf{y}'} = \Sigma_{\mathbf{y}}$  – i.e. the origin shift has no influence on the covariance of the data in the principal components axes, and can be used for convenience in displaying principal component images.

### 6.1.5

#### Application of Principal Components in Image Enhancement and Display

In constructing a colour display of remotely sensed data only three dimensions of information can be mapped to the three colour primaries of the display device. For imagery with more than three bands that means the user must choose the most appropriate subset of three to use. A less ad hoc means for colour assignment rests upon performing a principal components transform and assigning the first three components to the red, green and blue colour primaries.

Examination of a typical set of principal component images for Landsat data, such as those seen in Fig. 6.6, reveals that there is very little detail in the fourth component so that, in general, it could be ignored without prejudicing the ability to extract meaningful information from the scene. A difficulty with principal components colour display, however, is that there is no longer a one to one mapping between sensor wavelength bands and colours. Rather each colour now represents a linear combination of spectral components, making photointerpretation difficult for many applications. An exception would be in exploration geology where structural differences may be enhanced in principal components imagery, there often being little interest in the meanings of the actual colours.

### 6.1.6

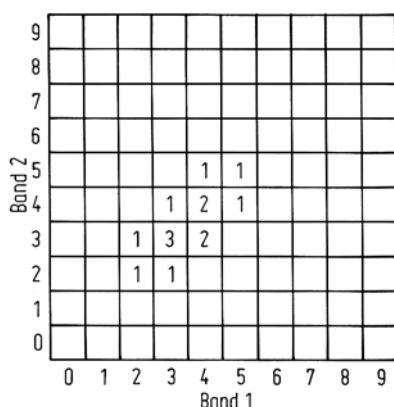
#### The Taylor Method of Contrast Enhancement

It will be demonstrated below that application of the contrast modification techniques of Chap. 4 to each of the individual components of a highly correlated vector image will yield an enhanced image in which certain highly saturated hues are missing. An interesting contrast stretching procedure which can be used to create a modified image with good utilisation of the range of available hues rests upon the use of the principal components transformation. It was developed by Taylor (1973) and has also been presented by Soha and Schwartz (1978). A more recent and general treatment has been given by Campbell (1996).

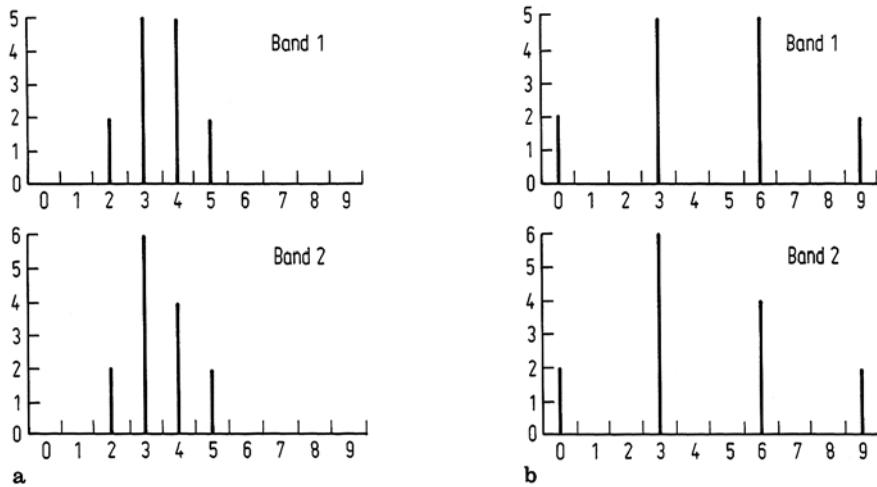
Consider a two dimensional image with the (two dimensional) histogram shown in Fig. 6.8. As observed the two components are highly correlated as revealed also from an inspection of the covariance matrix for the image which is

$$\Sigma_x = \begin{bmatrix} 0.885 & 0.616 \\ 0.616 & 0.879 \end{bmatrix} \quad (6.9)$$

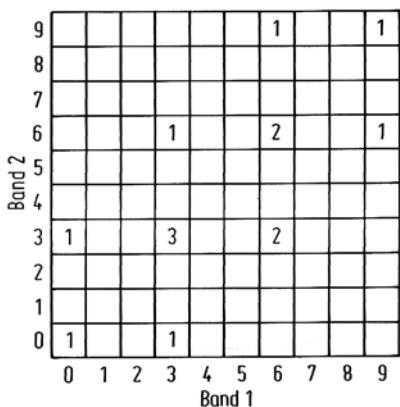
The range of brightness values occupied in the histogram suggests that there is value in performing a contrast stretch. Suppose a simple linear stretch is decided upon; the conventional means then for implementing such an enhancement with a multicomponent image is to apply it to each component independently. This requires the one dimensional histogram for each component to be constructed. These are obtained by counting the number of pixels with a given brightness value in each component, irrespective of their brightness in the other component – in other words they are marginal distributions of the two dimensional distribution. The single dimensional histograms corresponding to Fig. 6.8 are shown in Fig. 6.9a and the result of applying linear contrast enhancement to each of these is seen in Fig. 6.9b. The two dimensional histogram resulting from the contrast stretches applied to the individual components is shown in Fig. 6.10 wherein it is seen that the correlation between the components



**Fig. 6.8.** Histogram for a hypothetical two dimensional image showing correlation in its components. The numbers indicated on the bars (out of page) are the counts



**Fig. 6.9.** a Individual histograms for the image with the two dimensional histogram of Fig. 6.8;  
b The individual histograms after a simple linear contrast stretch over all available brightness values



**Fig. 6.10.** Histogram of a two dimensional image after simple linear contrast stretch of the components individually

is still present and that if component 1 is displayed as red and component 2 as green, no highly saturated reds or greens will be evident in the enhanced image, although brighter yellows will be more obvious than in the original data. It is a direct result of the correlation in the image that the highly saturated colour primaries are not displayed. The situation is even worse for display of three dimensional correlated image data. Simple contrast enhancement of each component independently will yield an image without highly saturated reds, blues and greens but also without saturated yellows, cyans and magentas. The procedure recommended by Taylor overcomes this, as demonstrated now. This fills the available colour space on the display more fully.

Let  $\mathbf{x}$  be the vector of brightness values of the pixels in the original image and  $\mathbf{y}$  be the corresponding vector of intensities after principal components transformation, such that  $\mathbf{y} = G\mathbf{x}$ .  $G$  is the principal components transformation matrix, composed of transposed eigenvectors of the original covariance matrix  $\Sigma_x$ . The covariance matrix which describes the scatter of pixel points in the principal components ( $\mathbf{y}$ ) vector space is a diagonal matrix of eigenvalues which, for three dimensional data, is of the form

$$\Sigma_y = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

Suppose now the individual principal components are enhanced in contrast such that they each cover the corresponding range of brightness values and, in addition, have the same variances; in other words the histograms of the principal components are matched, for example, to a Gaussian histogram that has the same variance in all dimensions. The new covariance matrix will therefore be of the form

$$\Sigma'_y = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix} = \sigma^2 \mathbf{I}$$

where  $\mathbf{I}$  is the identity matrix. Since the principal components are uncorrelated, enhancement of the components independently yields an image with good utilisation of the available colour space, with all hues possible. The axes in the colour space however are principal components axes and, as noted in the previous section, are not as desirable for photointerpretation as having a colour space based upon the original components of the image. It would be of particular value therefore if the image data could be returned to the original  $\mathbf{x}$  space to give a one-to-one mapping between the display colours and image components. Let the contrast enhanced principal components be represented by the vector  $\mathbf{y}'$ . These can be transformed back to the original axes for the image by using the inverse of the principal components transformation matrix  $G^{-1}$ . Since  $G$  is orthogonal its inverse is simply its transpose, which is readily available. The new covariance matrix of the data back in the original image domain is

$$\Sigma'_x = G^t \mathcal{E}\{(y' - \mathcal{E}(y'))(y' - \mathcal{E}(y'))^t\} G$$

where  $\mathbf{x}' = G^t \mathbf{y}'$ , is the modified pixel vector in the original space. Consequently

$$\begin{aligned} \Sigma'_x &= G^t \mathcal{E}\{(y' - \mathcal{E}(y'))(y' - \mathcal{E}(y'))^t\} G \\ &= G^t \Sigma'_y G \\ &= G^t \sigma^2 I G \end{aligned}$$

i.e.  $\Sigma'_x = \sigma^2 I$ .

Thus the covariance matrix of the enhanced principal components data is preserved on transformation back to the original image space. No correlation is introduced and the data shows good utilisation of the colour space using the original image data components. In practice, one problem encountered with the Taylor procedure is the noise introduced into the final results by the contrast enhanced third principal

component. Should all possible brightness values be available in the components this would not occur. However because most image analysis software treats image data in integer format in the range 0 to 255, rounding of intermediate results to integer form produces the noise. One possible remedy is to filter the noisy components before the inverse transform is carried out.

It will be appreciated from the foregoing discussion that colour composite principal component imagery will appear more colourful than a colour composite product formed from original image bands. This is a direct result of the ability to fill the colour space completely by contrast enhancing the uncorrelated components, by comparison to the poor utilization of colour by the original correlated data, as seen in the illustration of Fig. 6.10 and as demonstrated in Fig. 6.6c.

#### 6.1.7

#### Other Applications of Principal Components Analysis

Owing to the information compression properties of the principal components transformation it lends itself to reduced representation of image data for storage or transmission. In such a situation only the uppermost significant components are retained as a representation of an image, with the information content so lost being indicated by the sum of the eigenvalues corresponding to the components ignored. Thereafter if the original image is to be restored, either on reception through a communications channel or on retrieval from memory, then the inverse of the transformation matrix is used to reconstruct the image from the reduced set of components. Since the matrix is orthogonal its inverse is simply its transpose. This technique is known as bandwidth compression in the field of telecommunications. Until recently it had not found great application in satellite remote sensing image processing, because hitherto image transmission has not been a consideration and available memory has not placed stringent limits on image storage. With increasing use of imaging spectrometry data however (Sect. 1.2), bandwidth compression has become more important, as discussed in Sect. 13.8.

An interesting application of principal components analysis is in the detection of features that change with time between images of the same region. This is described by example in Chap. 11.

## 6.2

## Noise Adjusted Principal Components Transformation

In the example of Fig. 6.6 it is apparent that any noise present in the original image has been concentrated in the later principal components. Ordinarily that is what would be expected: ie. that the components would become progressively noisier as their eigenvalues decrease. In practice, however, that is not always the case. It is found, sometimes, that earlier components are noisier than those with the smallest eigenvalues. The noise adjusted transformation overcomes that problem (Lee et al, 1990).

Let

$$\mathbf{y} = \mathbf{G}\mathbf{x} = \mathbf{D}^t \mathbf{x} \quad (6.10)$$

be a transformation that will achieve what we want. As with the principal components transformation, if  $\Sigma_x$  is the covariance of the data in the original (as recorded) coordinate system, then the covariance matrix after transformation will be

$$\Sigma_y = \mathbf{D}^t \Sigma_x \mathbf{D} \quad (6.11)$$

To find the value for the transformation matrix  $\mathbf{D}^t$  that will order the noise by component we start by defining the *noise fraction*

$$\gamma = \frac{v^n}{v} \quad (6.12)$$

where  $v^n$  is the noise variance along a particular axis (i.e. in a given band) and  $v$  is the total variance along that axis (in that band), consisting of the sum of the signal (wanted) variance and the noise variance, assuming the signal and noise are uncorrelated. The total noise variance over all bands in the recorded data can be expressed as a noise covariance matrix  $\Sigma_x^n$  so that after transformation according to (6.10) the noise covariance matrix will be

$$\Sigma_y^n = \mathbf{D}^t \Sigma_x^n \mathbf{D} \quad (6.13)$$

Along one particular axis ( $\mathbf{g}$ ) the noise and total variances are then

$$\begin{aligned} v^n &= \mathbf{d}^t \Sigma_x^n \mathbf{d} \\ v &= \mathbf{d}^t \Sigma_x \mathbf{d} \end{aligned}$$

so that (6.12) becomes

$$\gamma = \frac{\mathbf{d}^t \Sigma_x^n \mathbf{d}}{\mathbf{d}^t \Sigma_x \mathbf{d}} \quad (6.14)$$

We now want to find that new coordinate direction  $\mathbf{g} = \mathbf{d}^t$  that minimises  $\gamma$ . To do so, we take the first derivative of  $\gamma$  with respect to  $\mathbf{d}$  zero.

Noting that  $\frac{\partial}{\partial \mathbf{d}} \{\mathbf{x}^t \mathbf{A} \mathbf{x}\} = 2\mathbf{A} \mathbf{x}$  then we have from (6.14)

$$\begin{aligned} \frac{\partial \gamma}{\partial \mathbf{d}} &= 2\Sigma_x^n \mathbf{d} \{\mathbf{d}^t \Sigma_x \mathbf{d}\}^{-1} - 2\Sigma_x \mathbf{d} \{\mathbf{d}^t \Sigma_x \mathbf{d}\}^{-2} \{\mathbf{d}^t \Sigma_x^n \mathbf{d}\} \\ &= 0 \end{aligned}$$

which, after simplification, leads to

$$\begin{aligned} \Sigma_x^n \mathbf{d} - \Sigma_x \mathbf{d} \frac{\mathbf{d}^t \Sigma_x^n \mathbf{d}}{\mathbf{d}^t \Sigma_x \mathbf{d}} &= 0 \\ \text{or } (\Sigma_x^n - \Sigma_x \gamma) \mathbf{d} &= 0 \\ \text{so that } (\Sigma_x^n \Sigma_x^{-1} - \gamma I) \mathbf{d} &= 0 \end{aligned} \quad (6.15)$$

Thus the  $\gamma$  are the eigenvalues of  $\Sigma_x^n \Sigma_x^{-1}$  and  $\mathbf{d}$  are the associated eigenvectors. If we rank the eigenvalues in *increasing* order, then the image components will be ranked from that with the lowest noise variance to that with the highest, as required.

Suppose now the noise covariance can be transformed to the identity matrix  $I$  (we will see how to do that below), then (6.15) becomes

$$(\Sigma_x^{-1} - \gamma I)\mathbf{d} = 0$$

Also, since  $\Sigma_x^n = I$  then  $\gamma = v^{-1}$ , so that the last expression can be written, after multiplying throughout by  $\Sigma_x$ ,

$$(\Sigma_x - vI)\mathbf{d} = 0$$

which is the standard eigenvalue equation associated with the usual principal components transformation. Note, as expected that  $v$ , the eigenvalue, is now explicitly the image variance in the relevant band, as might be expected. Therefore we now have a simple way to apply the noise adjusted principal components transformation – ie to ensure that the transformed images are ranked in increasing order of noise variance: first we transform the original data such that its noise covariance is the identity matrix, and then we apply the standard principal components procedure.

The only outstanding step is to know how to transform the data so it has a unity noise covariance. That can be achieved in the following manner.

From Appendix D we see that a diagonal form for  $\Sigma_x^n$  is

$$\Lambda = E^{-1} \Sigma_x^n E$$

in which  $\Lambda$  is the diagonal matrix of its eigenvalues and  $E$  is the matrix of its eigenvectors. However, we want the diagonal form to be the identity matrix. To generate that we pre-multiply the last expression by  $\Lambda^{-1/2t}$  and post-multiply it by  $\Lambda^{-1/2}$  so that we end up with

$$I = \Lambda^{(-1/2)t} E^{-1} \Sigma_x^n E \Lambda^{-1/2}$$

If we define  $F = E\Lambda^{-1/2}$ , so that

$$I = F^t \Sigma_x^n F$$

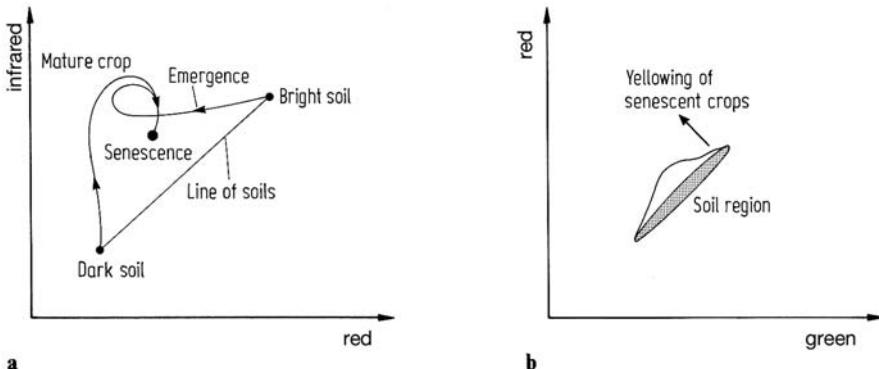
we recognise  $\mathbf{y} = F^t \mathbf{x}$  as the transformation of the original data that will yield a new data set in which the noise covariance matrix is unity. Provided this transformation is carried out first (which involves finding or estimating the noise covariance, and then finding its eigenvalues and eigenvectors) then the standard principal components transformation can be applied.

There are several ways the noise content of an image can be estimated. Many are based on examining the local properties of an image in segments thought to represent homogeneous regions on the ground. For those areas the residual data created by subtracting a smoothed version of the image from the original is assumed to represent noise. Olsen (1993) provides an overview of noise estimation methods.

## 6.3

### The Kauth-Thomas Tasseled Cap Transformation

The principal components transformation treated in Sects. 6.1 and 6.2 yields a new co-ordinate description of multispectral remote sensing image data by establishing a



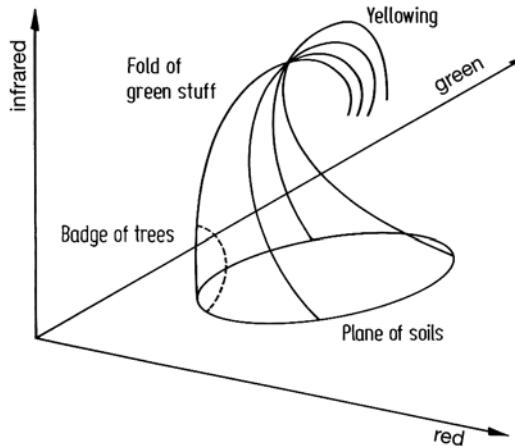
**Fig. 6.11.** **a** Infrared versus red subspace showing trajectories of crop development; **b** Red versus green subspace also depicting crop development

diagonal form of the global covariance matrix. The new co-ordinates (components) are linear combinations of the original spectral bands. Other linear transformations are of course possible. One is a procedure referred to as canonical analysis, treated in Chap. 10. Another, to be developed below, is application-specific in that the new axes in which data are described have been devised to maximise information of importance, in this case, to agriculture. Other similar special transformations would also be possible.

The so-called “tasseled cap” transformation (Crist and Kauth, 1986) developed by Kauth and Thomas (1976) is a means for highlighting the most important (spectrally observable) phenomena of crop development in a way that allows discrimination of specific crops, and crops from other vegetative cover, in Landsat multitemporal, multispectral imagery. Its basis originally lies in an observation of crop trajectories in band 6 versus band 5, and band 5 versus band 4 subspaces. Consider the former as shown in Fig. 6.11a.

A first observation that can be made is that the variety of soil types on which specific crops might be planted appear as points along a diagonal in an infrared, red space as shown. This is well-known and can be assessed from an observation of the spectral reflectance characteristics for soils. (See for example Chap. 5 of Swain and Davis, 1978.) Darker soils lie nearer the origin and lighter soils at higher values in both bands. The actual slope of this line of soils will depend upon global external variables such as atmospheric haze and soil moisture effects. If the transformation to be derived is to be used quantitatively these effects need to be modelled and the data calibrated or corrected beforehand.

Consider now the trajectories followed in infrared versus red subspace for crop pixels corresponding to growth on different soils – in this case take the extreme light and dark soils as depicted in Fig. 6.11a. For both regions at planting the multispectral response is dominated by soil types, as expected. As the crops emerge the shadows cast over the soil dominate any green matter response. As a result there is considerable darkening of the response of the lighter soil crop field and only a slight darkening



**Fig. 6.12.** Crop trajectories in a green, red, infrared space, having the appearance of a tasseled cap

of that on dark soil. When both crops reach maturity their trajectories come together implying closure of the crop canopy over the soil. The response is then dominated by the green biomass, being in a high infrared and low red region, as is well known. When the crops senesce and turn yellow their trajectories remain together and move away from the green biomass point in the manner depicted in the diagram. However whereas the development to maturity takes place almost totally in the same plane, the yellowing development in fact moves out of this plane, as can be assessed by how the trajectories develop in the red versus green subspace during senescence as illustrated in Fig. 6.11b.

Should the crops then be harvested, the trajectories beyond senescence move, in principle, back towards their original soil positions.

Having made these observations, the two diagrams of Fig. 6.11 can now be combined into a single three dimensional version in which the stages of the crop trajectories can be described according to the parts of a cap, with tassels, from which the name of the subsequent transformation is derived. This is shown in Fig. 6.12. The first point to note is that the line of soils used in Fig. 6.11a is shown now as a plane of soils. Its maximum spread is along the three dimensional diagonal as indicated; however it has a scatter about this line consistent with the spread in red versus green as shown in Fig. 6.11b. Kauth and Thomas note that this plane of soils forms the brim and base of the cap. As crops develop on any soil type their trajectories converge essentially towards the crown of the cap at maturity whereupon they fold over and continue to yellowing as indicated. Thereafter they break up to return ultimately to various soil positions, forming tassels on the cap as shown.

The behaviour observable in Fig. 6.12 led Kauth and Thomas to consider the development of a linear transformation that would be useful in crop discrimination. As with the principal components transform, this transformation will yield four orthogonal axes. However the axis directions are chosen according to the behaviour seen in Fig. 6.12.

Three major orthogonal directions of significance in agriculture can be identified. The first is the principal diagonal along which soils are distributed. This was chosen by Kauth and Thomas as the first axis in the tasseled cap transformation. The development of green biomass as crops move towards maturity appears to occur orthogonal to the soil major axis. This direction was then chosen as the second axis, with the intention of providing a greenness indicator. Crop yellowing takes place in a different plane to maturity. Consequently choosing a third axis orthogonal to the soil line and greenness axis will give a yellowness measure. Finally a fourth axis is required to account for data variance not substantially associated with differences in soil brightness or vegetative greenness or yellowness. Again this needs to be orthogonal to the previous three. It was called “non-such” by Kauth and Thomas in contrast to the names “soil brightness”, “green-stuff” and “yellow-stuff” they applied to the previous three.

The transformation that produces the new description of the data may be expressed as

$$\mathbf{u} = R\mathbf{x} + \mathbf{c} \quad (6.16)$$

where  $\mathbf{x}$  is the original Landsat vector, and  $\mathbf{u}$  is the vector of transformed brightness values. This has soil brightness as its first component, greenness as its second and yellowness as its third. These can therefore be used as indices, respectively.  $R$  is the transformation matrix and  $\mathbf{c}$  is a constant vector chosen (arbitrarily) to avoid negative values in  $\mathbf{u}$ .

The transformation matrix  $R$  is the transposed matrix of column unit vectors along each of the transformed axes (compare with the principal components transformation matrix). For a particular agricultural region Kauth and Thomas chose the first unit vector as a line of best fit through a set of soil classes. The subsequent unit vectors were generated by using a Gram-Schmidt orthogonalization procedure in the directions required. The transformation matrix generated for Landsat MSS data was

$$R = \begin{bmatrix} 0.433 & 0.632 & 0.586 & 0.264 \\ -0.290 & -0.562 & 0.600 & 0.491 \\ -0.829 & 0.522 & -0.039 & 0.194 \\ 0.223 & 0.012 & -0.543 & 0.810 \end{bmatrix}$$

From this it can be seen, at least for the region investigated by Kauth and Thomas, that the soil brightness is a weighted sum of the original four Landsat bands with approximately equal emphasis. The greenness measure is the difference between the infrared and visible responses. In a sense therefore this is more a biomass index. The yellowness measure can be seen to be substantially the difference between the Landsat visible red and green bands.

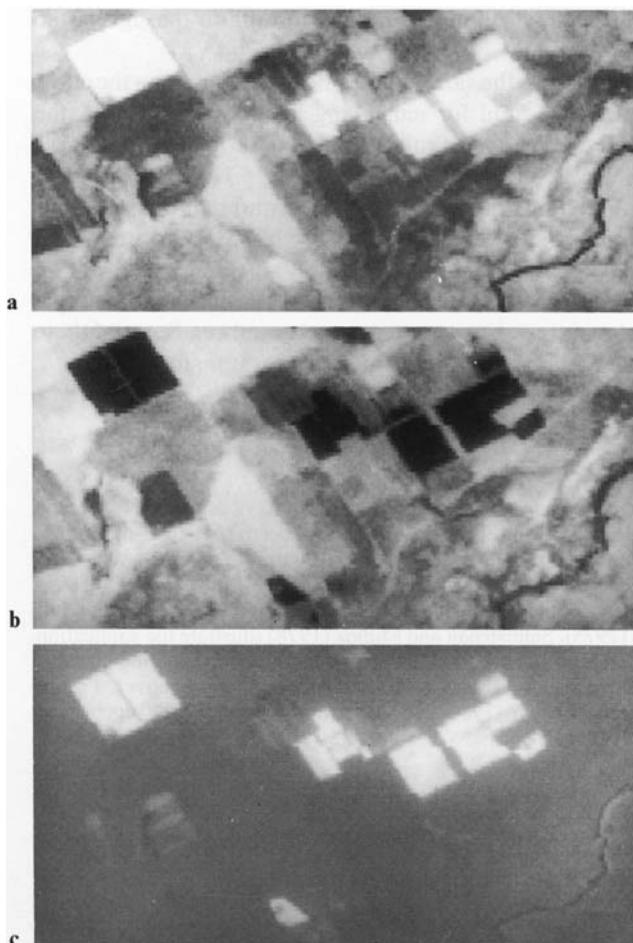
Just as new images can be synthesised to correspond to various principal components so can the actual transformed images be created for this approach. By applying (6.16) to every pixel in a Landsat multispectral scanner image, soil brightness, greenness, yellowness and non-such images can be produced. These can then be used to assess stages in crop development. The method can also be applied to other sensors.

## 6.4

### Image Arithmetic, Band Ratios and Vegetation Indices

Addition, subtraction, multiplication and division of the pixel brightnesses from two bands of image data to form a new image are particularly simple transformations to apply. Multiplication seems not to be as useful as the others, band differences and ratios being most common.

Differences can be used to highlight regions of change between two images of the same area. This requires that the images be registered using the techniques of Chap. 2 beforehand. The resultant difference image must be scaled to remove



**Fig. 6.13.** Landsat multispectral scanner band 7 **a** and band 5, **b** images of an arid region containing irrigated crop fields. The ratio of these two images **c** shows vegetated regions as bright, soils as mid to dark grey and water as black

negative brightness values. Normally this is done so that regions of no change appear mid-grey, with changes shown as brighter or duller than mid-grey according to the sign of the difference.

Ratios of different spectral bands from the same image find use in reducing the effect of topography, as a vegetation index, and for enhancing subtle differences in the spectral reflectance characteristics for rocks and soils. As an illustration of the value of band ratios for providing a single vegetation index image, Fig. 6.13 shows Landsat multispectral scanner band 5 and band 7 images of an agricultural region along with the band 7/band 5 ratio. As seen, healthy vegetated areas are bright, soils are mid to dark grey, and water is black. These shades are readily understood from an examination of the corresponding spectral reflectance curves. Variations on simple arithmetic operations between bands are also sometimes used as indices. Some of these are treated in Sect. 10.4.6. Note that band ratioing is not a linear transformation.

## References for Chapter 6

An easily read treatment of the principal components transformation has been given by Jensen and Waltz (1979), although the degree of mathematical detail has been kept to a minimum. Theoretical treatments can be found in many books on pattern recognition, image analysis and data analysis, although often under the alternative titles of Karhunen-Loëve and Hotelling transforms. Treatments of this type that could be consulted include Andrews (1972), Gonzalez and Woods (1992) and Ahmed and Rao (1975). Santisteban and Muñoz (1978) illustrate the application of the technique. The transformation has also been looked at as a method for detecting changes between successive images of the same region. This is illustrated in Sect. 11.7 and covered more fully in the papers by Byrne, Crapper and Mayo (1980), Howarth and Boasson (1983), Ingebritsen and Lyon (1985) and Richards (1984).

- N. Ahmed and K.R. Rao, 1975: Orthogonal Transforms for Digital Signal Processing, Berlin, Springer-Verlag
- H.C. Andrews, 1972: Introduction to Mathematical Techniques in Pattern Recognition, New York, Wiley.
- E.F. Byrne, P.F. Crapper and K.K. Mayo, 1980: Monitoring Land-Cover Change by Principal Components Analysis of Multitemporal Landsat Data. *Remote Sensing of Environment*, 10, 175–184.
- N.A. Campbell, 1996: The Decorrelation Stretch Transformation. *Int. J. Remote Sensing*, 17, 1939–1949.
- E.P. Crist and R.T. Kauth, 1986: The Tasseled Cap De-Mystified. *Photogrammetric Engineering and Remote Sensing*, 52, 81–86.
- R.C. Gonzalez and R.E. Woods, 1992: Digital Image Processing, Mass., Addison-Wesley.
- P.J. Howarth and E. Boasson, 1983: Landsat Digital Enhancements for Change Detection in Urban Environments. *Remote Sensing of Environment*, 13, 149–160.
- S.E. Ingebritsen and R.J.P. Lyon, 1985: Principal Components Analysis of Multitemporal Image Pairs. *Int. J. Remote Sensing*, 6, 687–696.
- S.K. Jensen and F.A. Waltz, 1979: Principal Components Analysis and Canonical Analysis in Remote Sensing. *Proc. American Photogrammetric Soc. 45th Ann. Meeting*, 337–348.

- R.J. Kauth and G.S. Thomas, 1976: The Tasseled Cap – A Graphic Description of the Spectral-Temporal Development of Agricultural Crops as Seen by Landsat. Proc. LARS 1976 Symp. on Machine Process. Remotely Sensed Data, Purdue University.
- S.I. Olsen, 1993: Estimation of Noise in Images: an Evaluation. Graphical Models and Image Processing, 55, 319–323.
- J.A. Richards, 1984: Thematic Mapping from Multitemporal Image Data Using the Principal Components Transformation. Remote Sensing of Environment, 16, 35–46.
- A. Santisteban and L. Muñoz, 1978: Principal Components of a Multispectral Image: Application to a Geologic Problem. IBM J. Research and Development, 22, 444–454.
- J.M. Soha and A.A. Schwartz, 1978: Multispectral Histogram Normalization Contrast Enhancement. Proc. 5th Canadian Symp. on Remote Sensing, 86–93.
- P.H. Swain and S.M. Davis (Eds.), 1978: Remote Sensing: The Quantitative Approach, New York, McGraw-Hill.
- M.M. Taylor, 1973: Principal Components Colour Display of ERTS Imagery. Third Earth Resources Technology Satellite-1 Symposium, NASA SP-351, 1877–1897.

## Problems

**6.1** (a) At a conference research group A and research group B both presented papers on the value of the principal components transformation (also known as the Karhunen-Loëve or Hotelling transform) for reducing the number of features required to represent image data. Group A described very good results that they had obtained with the method whereas Group B indicated that they felt it was of little use. Both groups were using image data with only two spectral components. The covariance matrices for their respective images are:

$$\Sigma_A = \begin{bmatrix} 5.4 & 4.5 \\ 4.5 & 6.1 \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 28.0 & 4.2 \\ 4.2 & 16.4 \end{bmatrix}$$

Explain the points of view of both groups.

(b) If information content can be related directly to variance indicate how much information is discarded if only the first principal component is retained by both groups.

**6.2** Suppose you have been asked to describe the principal components transformation to a non-specialist. Write a single paragraph summary of its essential features, using diagrams if you wish, but no mathematics.

**6.3** (For those mathematically inclined), Demonstrate that the principal components transformation matrix developed in Sect. 6.1.2 is orthogonal.

**6.4** Colour image products formed from principal components generally appear richer in colour than a colour composite product formed by combining the original bands of remote sensing image data. Why do you think that is so?

**6.5** (a) The steps involved in computing principal component images may be summarised as:

- calculation of the image covariance matrix
- eigenanalysis of the covariance matrix
- computation of the principal components.

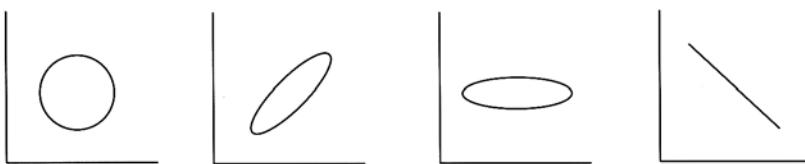
Assessments can be made in the first two steps as to the likely value in proceeding to compute the components. Describe what you would look for in each case.

(b) The covariance matrix need not be computed over the full image to produce a principal

components transformation. Discuss the value of using training areas to define the portion of image data to be taken into account in compiling the covariance matrix.

**6.6** Imagine you have two images from a sensor which has a single band in the range 0.9 to 1.1  $\mu\text{m}$ . One image was taken before a flood occurred. The second shows the extent of flood inundation. Produce a sketch of what the “two-date” multispectral space would look like if the image from the first date contained rich vegetation, sand and water and that in the second date contains the same cover types but with an expanded region of water. Demonstrate how a two dimensional principal components transform can be used to highlight the extent of flooding.

**6.7** Describe the nature of the correlations between the pairs of axis variables (e.g. bands) in each of the cases in Fig. 6.14.



**Fig. 6.14.** Examples of two dimensional correlations

**6.8** The covariance matrix for an image recorded by a particular four channel sensor is as shown below. Which band would you discard if you had to construct a colour composite display of the image by assigning the remaining three bands to each of the colour primaries?

$$\Sigma = \begin{bmatrix} 35 & 10 & 10 & 5 \\ 10 & 20 & 12 & 2 \\ 10 & 12 & 40 & 30 \\ 5 & 2 & 30 & 30 \end{bmatrix}$$

# 7

## Fourier Transformation of Image Data

### 7.1

#### Introduction

Many of the geometric enhancement techniques used with remote sensing image data can be carried out using the simple template-based techniques of Chap. 5. More flexibility is offered however if procedures are implemented in the so-called spatial frequency domain by means of the Fourier transformation. As a simple illustration, filters can be designed to extract periodic noise from an image that is unable to be removed by practical templates. As demonstrated in Sect. 5.4 the computational cost of using Fourier transformation for geometric operations is high by comparison to the template methods usually employed. However with the computational capacity of modern workstations, and the flexibility available in Fourier transform processing, this approach is one that should not be ignored.

Development of Fourier transform theory depends upon a knowledge of complex numbers and facility with integral calculus. The reader without that background may wish to pass over this Chapter and may do so without detracting from material in the remainder of the book. It is the purpose of the Chapter to present an overview of the significant aspects of the theory of Fourier transformation of image data. In its entirety the topic is an extensive one and well beyond the scope of this treatment. Instead the material presented in the following will serve to introduce the operational aspects of the topic, with little dependence on proofs and theory. Should the treatment be found to be too brief, particularly in the background material of Sects. 7.2 to 7.5, more details can be found in Brigham (1974, 1988), and McGillem and Cooper (1984).

Another transformation that now finds wide application to images is that based on the definition of wavelets (Castleman, 1996).

### 7.2

#### Special Functions

A number of mathematical functions are important in both developing and understanding the Fourier transformation. These are reviewed in this section along with some properties that will be of use later on.

Although functions of interest in image processing have position as their independent variable, it will be convenient here to use functions of time. These will be interpreted as functions of position as required.

### 7.2.1

#### The Complex Exponential Function

The complex exponential is defined by

$$f(t) = Re^{j\omega t} \quad (7.1a)$$

where  $j = \sqrt{-1}$ ,  $R$  is the amplitude of the function and  $\omega$  is called its radian frequency. The units of  $\omega$  are radians per second (or radians per unit of spatial variable). Frequently  $\omega$  is expressed in terms of “natural” frequency

$$f = \omega / 2\pi \quad (7.1b)$$

where  $f$  has units of hertz (or cycles per spatial variable). The complex exponential is periodic, with period  $T = 2\pi/\omega$ . This is appreciated by plotting it as a function of the independent variable on the complex (argand) plane. Alternatively, we can express

$$f(t) = Re^{\pm j\omega t} = R \cos \omega t \pm jR \sin \omega t \quad (7.1c)$$

to see its periodic behaviour in terms of sinusoids. For convenience we will now choose  $R = 1$ . From this last expression we see

$$\cos \omega t = \Re{e^{j\omega t}}$$

$$\sin \omega t = \Im{e^{j\omega t}}$$

where  $\Re$  and  $\Im$  are operators that select the real and imaginary parts of a complex number.

Finally, it can be seen from (7.1c)

$$\cos \omega t = \frac{1}{2}(e^{j\omega t} + e^{-j\omega t}) \quad (7.2a)$$

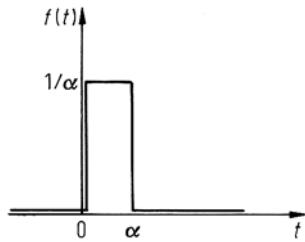
$$\sin \omega t = \frac{1}{2j}(e^{j\omega t} - e^{-j\omega t}) \quad (7.2b)$$

### 7.2.2

#### The Dirac Delta Function

A function of particular importance in determining properties of sampled signals, which include digital image data, is the impulse function, also referred to as the Dirac delta function. This is a spike-like function of infinite amplitude and infinitesimal duration. It cannot be defined explicitly. Instead it is defined by a limiting operation as in the following manner.

Consider the rectangular pulse of duration  $\alpha$  and amplitude  $1/\alpha$  as seen in Fig. 7.1. Note that the area under the curve is 1. Accordingly the delta function  $\delta(t)$  is defined



**Fig. 7.1.** Pulse which approaches an impulse in the limit as  $\alpha \rightarrow 0$

as the pulse in the limit as  $\alpha$  goes to zero. As a formal definition, the best that can be done is

$$\delta(t) = 0 \quad \text{for } t \neq 0 \quad (7.3a)$$

and

$$\int_{-\infty}^{\infty} \delta(t) dt = 1 \quad (7.3b)$$

This turns out to be sufficient for our purposes. Equation (7.3a) defines a delta function at the origin; an impulse at time  $t_0$  is defined by

$$\delta(t - t_0) = 0 \quad \text{for } t \neq t_0 \quad (7.4a)$$

and

$$\int_{-\infty}^{\infty} \delta(t - t_0) dt = 1 \quad (7.4b)$$

### 7.2.2.1

#### Properties of the Delta Function

From the definition of the delta function it can be seen that the product of a delta function with another function is

$$\delta(t - t_0) f(t) = \delta(t - t_0) f(t_0), \quad (7.5a)$$

from which we can see

$$\begin{aligned} \int_{-\infty}^{\infty} \delta(t - t_0) f(t) dt &= \int_{-\infty}^{\infty} \delta(t - t_0) f(t_0) dt \\ &= f(t_0) \int_{-\infty}^{\infty} \delta(t - t_0) dt \end{aligned}$$

i.e.

$$\int_{-\infty}^{\infty} \delta(t - t_0) f(t) dt = f(t_0) \quad (7.5b)$$

This is known as the sifting property of the impulse.

### 7.2.3

#### The Heaviside Step Function

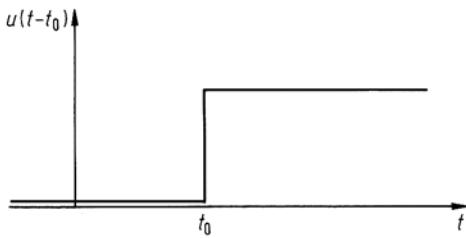
Figure 7.2 shows the Heaviside step function defined by

$$u(t - t_0) = 1 \quad \text{for } t \geq t_0 \quad (7.6a)$$

$$= 0 \quad \text{for } t < t_0 \quad (7.6b)$$

Note that it is 1 when its argument is zero or positive, and is zero for a negative argument. It can be seen that  $u(t)$  is related to  $\delta(t)$  by

$$\delta(t) = \frac{du(t)}{dt}$$



**Fig. 7.2.** The Heaviside step function

## 7.3

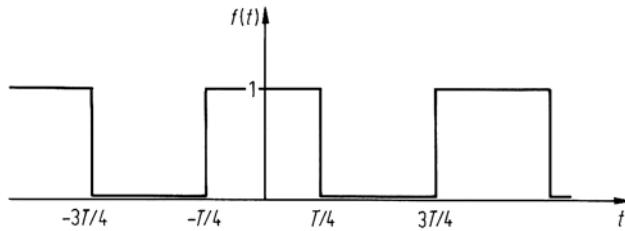
### Fourier Series

If a function  $f(t)$  is periodic with period  $T$  – i.e.  $f(t) = f(t + T)$  – then it can be expressed as an infinite sum of complex exponentials in the manner

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{jn\omega_0 t}, \quad \omega_0 = \frac{2\pi}{T} \quad (7.7a)$$

in which  $n$  is an integer and the complex expansion coefficients  $F_n$  are given by

$$F_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j n \omega_0 t} dt \quad (7.7b)$$



**Fig. 7.3.** A square waveform

The expressions in (7.7) are referred to as the exponential form of the Fourier series; (7.1c) also allows a trigonometric expression to be derived (McGillem and Cooper, 1984). Although (7.7a) is expressed in exponentials we often colloquially talk of (7.7a) as showing the sinusoidal spectral composition of  $f(t)$ . Equation (7.2) shows that this is acceptable and quite accurate.

As an illustration consider the need to determine the Fourier series of the square waveform in Fig. 7.3. From (7.7b) it can be seen that

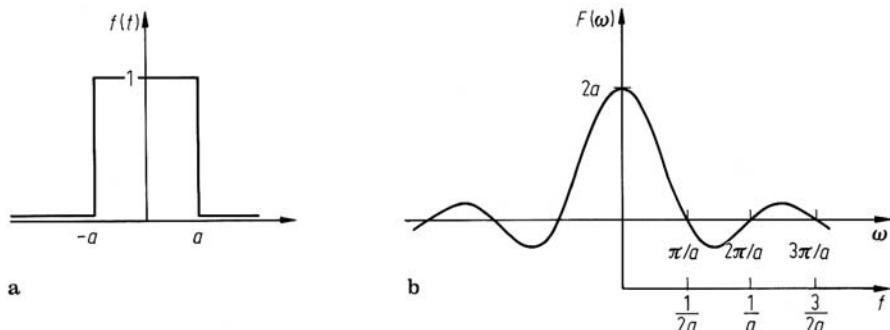
$$\begin{aligned} F_n &= \frac{1}{T} \int_{-T/4}^{T/4} e^{-jn\omega_0 t} dt \\ &= \frac{1}{n\pi} \sin \frac{n\pi}{2}. \end{aligned}$$

This tells the amount of each of the constituent  $e^{jn\omega_0 t}$  in (7.7a) required to represent the square waveform – i.e. it describes its sinusoidal composition. Note that when  $n = 0$ ,  $F_0 = 1/2$  as expected from Fig. 7.3. For  $n > 1$  the coefficients decrease in amplitude according to  $1/n$ . In general the  $F_n$  are complex and thus can be expressed in the form of an amplitude and phase, referred to respectively as amplitude and phase spectra.

## 7.4 The Fourier Transform

The Fourier series of the preceding section is a description of a periodic function in terms of a sum of sinusoidal terms (expressed in complex exponentials) at integral multiples of the so-called fundamental frequency  $\omega_0$ . For functions that are non-periodic, or *aperiodic* as they are sometimes called, decomposition into sinusoidal components requires use of the Fourier transformation. The transform itself, which is equivalent to the Fourier series coefficients of (7.7b), is defined by

$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (7.8a)$$



**Fig. 7.4.** a Unit pulse and b its Fourier transform

In general, an aperiodic function requires a continuum of sinusoidal frequency components for a Fourier description. Indeed if we plot  $F(\omega)$ , or for that matter its amplitude and phase, as a function of frequency it will be a continuous function of  $\omega$ . The function  $f(t)$  can be reconstructed from the spectrum according to

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (7.8b)$$

A Fourier transform of some importance is that of the unit pulse shown in Fig. 7.4a. From (7.8a) this is seen to be

$$F(\omega) = \int_{-a}^a e^{-j\omega t} dt = 2a \frac{\sin a\omega}{a\omega}$$

which is shown plotted in Fig. 7.4b. Note that the frequency axis accommodates both positive and negative frequencies. The latter have no physical meaning but rather are an outcome of using complex exponentials in (7.8a) instead of sinusoids.

It is also of interest to note the Fourier transform of an impulse

$$F(\omega) = \int_{-\infty}^{\infty} \delta(t) e^{-j\omega t} dt = 1$$

from the sifting property of the impulse (7.5b); the Fourier transform of a constant is

$$F(\omega) = \int_{-\infty}^{\infty} c e^{-j\omega t} dt = 2\pi c \delta(\omega).$$

This result is easily shown by working from the spectrum  $F(\omega)$  to the time function and again using the sifting property. In a like manner it can be shown that the Fourier transform of a periodic function is given by

$$F(\omega) = 2\pi \sum_{n=-\infty}^{\infty} F_n \delta(\omega - n\omega_0)$$

where  $F_n$  is the Fourier series coefficient corresponding to the frequency  $n\omega_0$ .

## 7.5 Convolution

### 7.5.1 The Convolution Integral

In Sect. 5.3 the concept of convolution was introduced as a means for determining the response of a linear system. It is also a very useful signal synthesis operation in general and finds particular application in the description of digital data, as will be seen in later sections. Here we express the convolution of two functions  $f_1(t)$  and  $f_2(t)$  as

$$y(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t - \tau) d\tau \triangleq f_1(t) * f_2(t) \quad (7.9)$$

It is a commutative operation, i.e.  $f_1(t) * f_2(t) = f_2(t) * f_1(t)$  a fact that can sometimes be exploited in evaluating the integral.

The convolution operation can be illustrated by interpreting the defining integral as representing the following four operations:

- (i) folding – form  $f_2(-\tau)$  by taking its mirror image about the ordinate axis
- (ii) shifting – form  $f_2(t - \tau)$  by shifting  $f_2(-\tau)$  by the amount  $t$
- (iii) multiplication – form  $f_1(\tau) f_2(t - \tau)$
- (iv) integration – compute the area under the product.

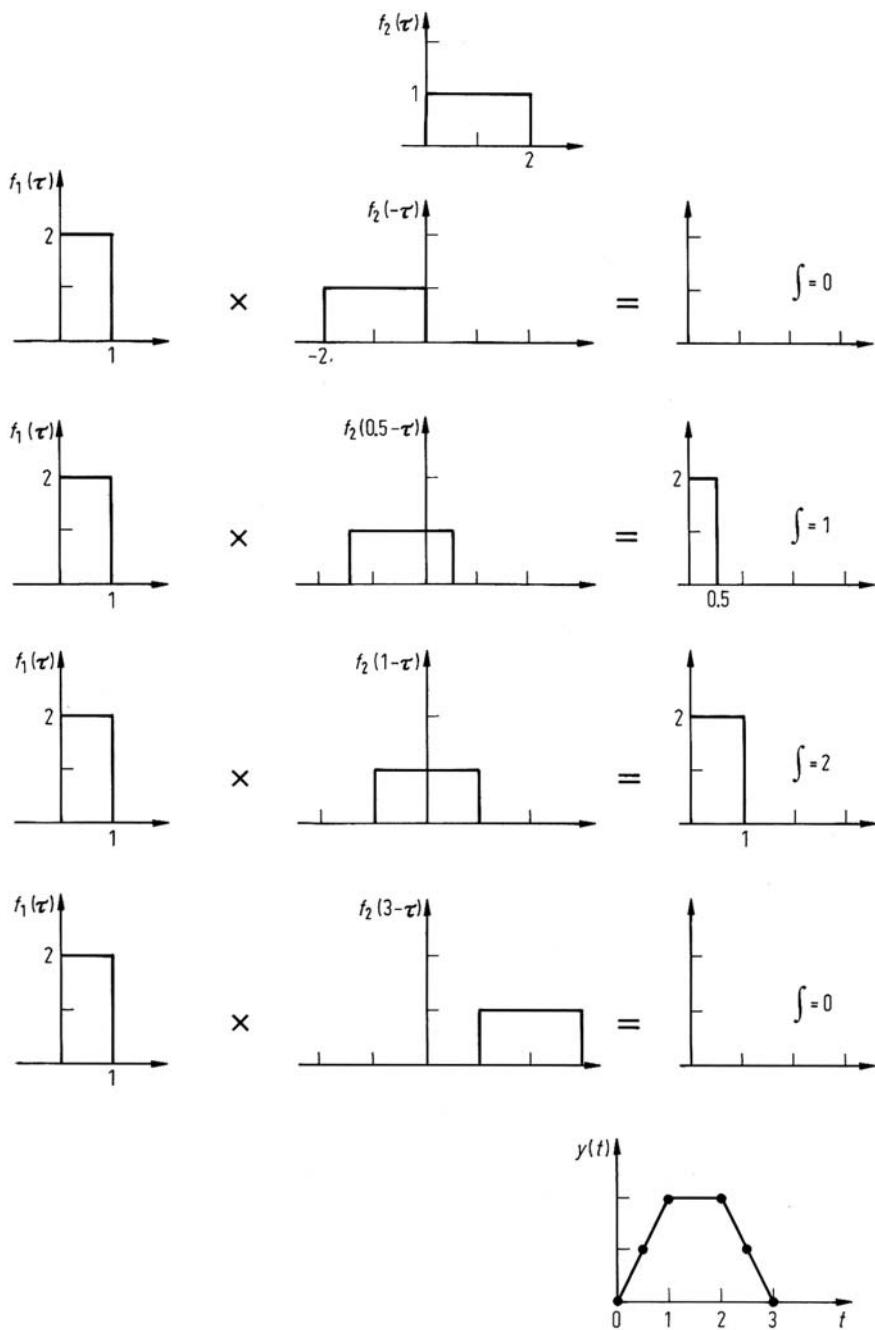
These steps are illustrated in Fig. 7.5.

### 7.5.2 Convolution with an Impulse

Convolution of a function with an impulse is important in sampling. The sifting theorem for the delta function, along with (7.9), shows

$$\begin{aligned} f(t) * \delta(t - t_0) &= \int_{-\infty}^{\infty} f(\tau) \delta(t - \tau - t_0) d\tau \\ &= f(t - t_0). \end{aligned}$$

Thus the effect is to shift the function  $f(t)$  to a new origin.

**Fig. 7.5.** Graphical illustration of the convolution operation

### 7.5.3

#### The Convolution Theorem

This theorem is readily verified using the definition of convolution and the definition of the Fourier transform. It has two forms (Papoulis, 1980). These are:

If

$$y(t) = f_1(t) * f_2(t)$$

then

$$Y(\omega) = F_1(\omega)F_2(\omega), \quad (7.10a)$$

and, if

$$Y(\omega) = F_1(\omega) * F_2(\omega)$$

then

$$y(t) = \frac{1}{2\pi} f_1(t)f_2(t) \quad (7.10b)$$

## 7.6

### Sampling Theory

The previous sections have dealt with functions that are continuous with time (or with position, as the case may be). However our interest principally is in functions, and images, that are discrete with time or position. Discrete time functions and digital images can be considered to be the result of the corresponding continuous functions having been sampled on a regular basis. Again, we will develop the concepts of sampling using functions of a single variable, such as time; the concepts are readily extended to two dimensional image functions.

A periodic sequence of impulses, spaced  $T$  apart,

$$\Delta(t) = \sum_{k=-\infty}^{\infty} \delta(t - kT) \quad (7.11)$$

can be considered as a sampling function, i.e. it can be used to extract a uniform set of samples from a function  $f(t)$  by forming the product

$$f = f(t)\Delta(t). \quad (7.12)$$

According to (7.5a),  $f$  is a sequence of samples of value  $f(kT) \delta(t - kT)$ . Despite the undefined magnitude of the delta function we will be content in this treatment to regard that product as a sample of the function  $f(t)$ . Strictly this should be interpreted in terms of so-called distribution theory; a simple interpretation of (7.12) as a set of uniformly spaced samples of  $f(t)$  however will not compromise our subsequent development.

It is important to consider the Fourier transform of the set of samples in (7.12) so that the frequency composition of a sampled function can be appreciated. This can be done using the convolution theorem (7.10b) provided the Fourier transform of  $\Delta(t)$  can be found.

The Fourier transform of  $\Delta(t)$  can be determined via its Fourier series. From (7.7b) and (7.5b) the Fourier series coefficients of  $\Delta(t)$  are given by

$$\Delta_n = \frac{1}{T} \int_{-T/2}^{T/2} \delta(t) e^{-jn\omega_0 t} dt = \frac{1}{T}$$

which, with the expression for the Fourier transform of a periodic function in Sect. 7.4, gives the Fourier transform of  $\Delta(t)$  as

$$\Delta(\omega) = \frac{2\pi}{T} \sum_{n=-\infty}^{\infty} \delta(\omega - n\omega_s) \quad (7.13)$$

where  $\omega_s = 2\pi/T$ . Thus the Fourier transform of the periodic sequence of impulses spaced  $T$  apart in time is itself a periodic sequence of impulses in the frequency domain, spaced  $2\pi/T$  rad s<sup>-1</sup> apart (or  $1/THz$  apart). Thus if  $f(t)$  has the spectrum  $F(\omega)$  (i.e. Fourier Transform) depicted in Fig. 7.6a then the spectrum of the set of samples in (7.12) is as shown in Fig. 7.6c. This is given by convolving  $F(\omega)$  with the sequence of impulses in (7.13), according to (7.10b). Recall that convolution with an impulse shifts a function to a new origin centred on the impulse.

Figure 7.6c demonstrates that the spectrum of a sampled function is a periodic repetition of the spectrum of the unsampled function, with the repetition period in the frequency domain determined by the rate at which the time function is sampled. If the sampling rate is high then the segments of the spectrum are well separated. If the sampling rate is low then the segments in the spectrum are close together.

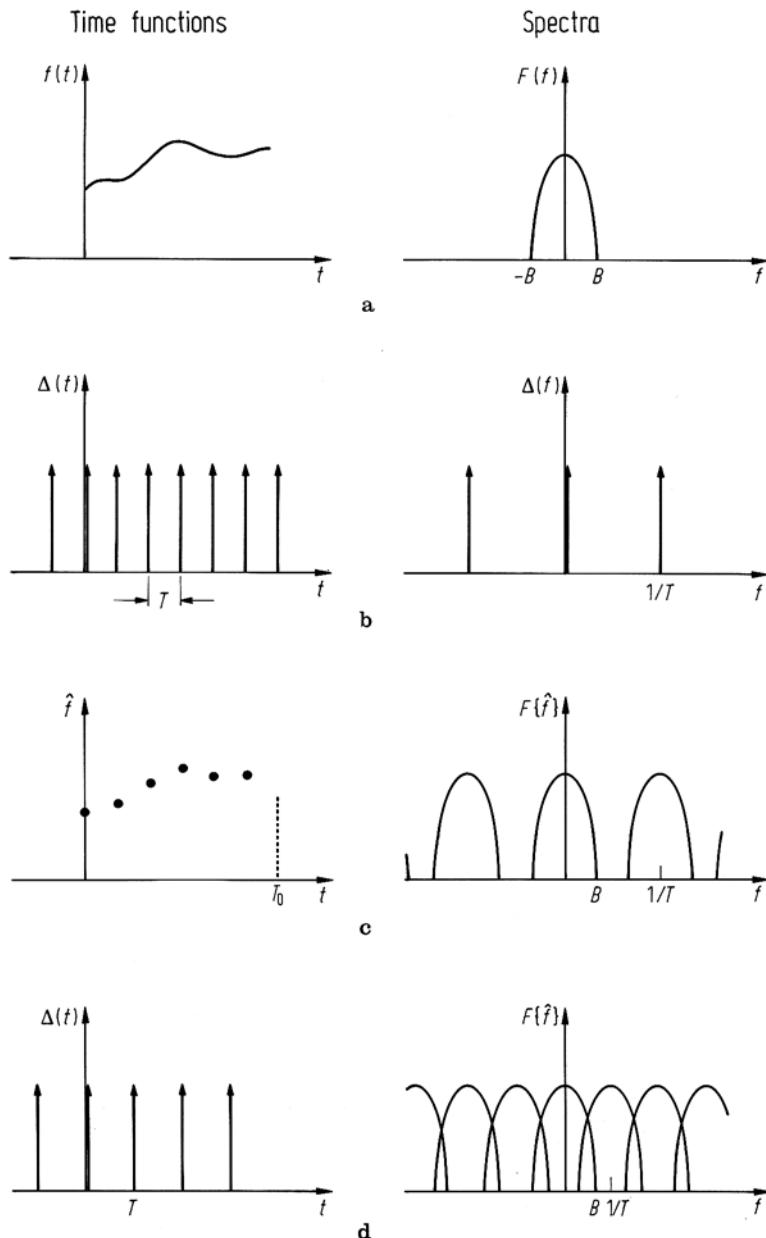
In the illustration shown in Fig. 7.6 the spectrum of  $f(t)$  is shown to be limited to frequencies below  $B$  Hz. ( $2\pi B$  rad · s<sup>-1</sup>);  $B$  is referred to as the bandwidth of  $f(t)$ . Not all real non-periodic functions have a limited bandwidth – the single pulse of Fig. 7.4 is an example of this – however it suits our purpose here to assume there is a limit to the frequency composition of functions of interest to us, defined by the signal bandwidth.

If adjacent segments are to remain separated as depicted in Fig. 7.6c then it is clear that

$$\frac{1}{T} > 2B \quad (7.14)$$

i.e. that the rate at which the function  $f(t)$  is sampled must exceed twice the bandwidth of  $f(t)$ . Should this not be the case then the segments of the spectrum of the sampled function overlap as shown in Fig. 7.6d, causing a form of distortion called *aliasing*.

A sampling rate of  $2 B$  in (7.14) is referred to as the Nyquist rate; Eq. (7.14) itself is often referred to as the *sampling theorem*.



**Fig. 7.6.** Development of the Fourier transform of a sampled function. **a** Unsampled function and its spectrum; **b** Periodic sequence of impulses and its spectrum; **c** Sampled function and its spectrum; **d** Sub-Nyquist rate sampling impulses and spectrum with aliasing.  $F$  represents Fourier transformation

## 7.7

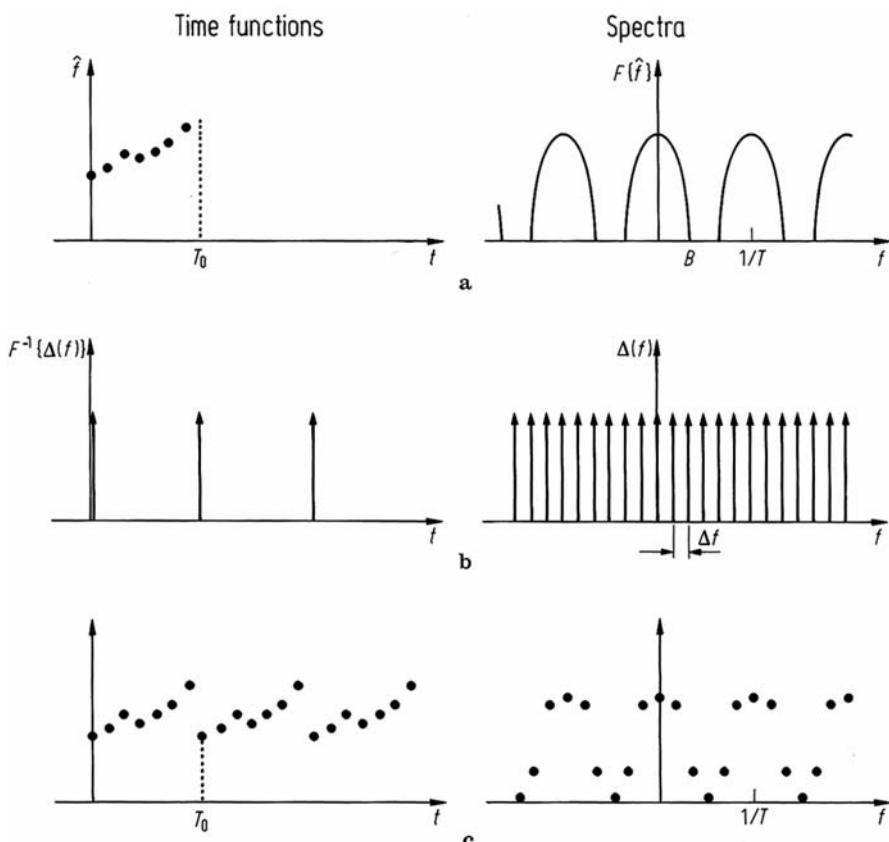
### The Discrete Fourier Transform

#### 7.7.1

#### The Discrete Spectrum

Consider now the problem of finding the spectrum (i.e. of computing the Fourier transform) of a sequence of samples. This is the first stage in our computation of the Fourier transform of an image. Indeed, the sequence of samples to be considered here could be looked at as a single line of pixels in digital image data.

Figure 7.7a shows that the spectrum of a set of samples is itself a continuous function of frequency. For digital processing clearly it is necessary that the spectrum be also represented by a set of samples, that would, for example, exist in computer



**Fig. 7.7.** Effect of sampling the spectrum. **a** Sampled function and its spectrum; **b** Periodic sequence of impulses used to sample the spectrum (right) and its time domain equivalent (left); **c** Sampled version of the spectrum (right) and its time domain equivalent (left); the latter is a periodic version of the samples in **a**. In these  $F^{-1}$  represents an inverse Fourier transformation

memory. Therefore we have to introduce a suitable sampling function also in the frequency domain. For this purpose consider an infinite periodic sequence of impulses in the frequency domain spaced  $\Delta f$  (i.e.  $\Delta\omega/2\pi$ ) apart as shown in Fig. 7.7b. It can be shown that the inverse transform of this sequence is another sequence of impulses in the time domain, spaced  $T_0 = 1/\Delta f$  apart. This can be appreciated readily from (7.11) and (7.13), although here we are going from the frequency domain to the time domain rather than vice versa.

If the (periodic) spectrum  $F(\omega)$  in Fig. 7.7a is multiplied by the frequency domain sampling function of Fig. 7.7b then the convolution theorem (7.10a) implies that the samples of  $f(t)$  will be formed into a periodic sequence with period  $T_0$  as illustrated in Fig. 7.7c. It is convenient if the number of samples used to represent the spectrum is the same as the actual number of samples taken of  $f(t)$ . Let this number be  $K$ . (There is a distortion introduced by using a finite rather than infinite number of samples. This will be addressed later.) Since the time domain has samples spaced  $T$  apart, the duration of sampling is  $KT$  seconds. It is pointless sampling the time domain over a period longer than  $T_0$  since no new information is added. Simply other periods are added. Consequently the optimum sampling time is  $T_0$ , so that  $T_0 = KT$ . Thus the sampling increment in the frequency domain is  $\Delta f = 1/T_0 = 1/KT$ . It is the inverse of the sampling duration. Likewise the total unambiguous bandwidth in the frequency domain is  $K \times \Delta f = 1/T$ , covering just one segment of the spectrum.

With those parameters established we can now consider how the Fourier transform operation can be modified to handle digital data.

### 7.7.2 Discrete Fourier Transform Formulae

Let the sequence  $\phi(k)$ ,  $k = 0, \dots, K - 1$  be the set of  $K$  samples taken of  $f(t)$  over the sampling period 0 to  $T_0$ . The samples correspond to times  $t_k = kT$ .

Let the sequence  $F(r)$ ,  $r = 0, \dots, K - 1$  be the set of samples of the frequency spectrum. These can be derived from the  $\phi(k)$  by suitably modifying (7.8a). For example, the integral over time can be replaced by the sum over  $k = 0$  to  $K - 1$ , with  $dt$  replaced by  $T$ , the sampling increment. The continuous function  $f(t)$  is replaced by the samples  $\phi(k)$  and  $\omega = 2\pi f$  is replaced by  $2\pi r \Delta f$ , with  $r = 0, \dots, K - 1$ . Thus  $\omega = 2\pi r/T_0$ . The time variable  $t$  is replaced by  $kT = kT_0/K$ ,  $k = 0, \dots, K - 1$ . With these changes (7.8a) can be written in sampled form as

$$F(r) = T \sum_{k=0}^{K-1} \phi(k) W^{rk}, \quad r = 0, \dots, K - 1 \quad (7.15)$$

with

$$W = e^{-j2\pi/K}. \quad (7.16)$$

Equation (7.15) is known as the *discrete Fourier transform* (DFT). In a similar manner a *discrete inverse Fourier transform* (DIFT) can be derived that allows reconstruction

of the time sequence  $\phi(k)$  from the frequency samples  $F(r)$ . This is

$$\phi(k) = \frac{1}{T_0} \sum_{r=0}^{K-1} F(r) W^{-rk}, k = 0, \dots, K-1 \quad (7.17)$$

Substitution of (7.15) into (7.17) shows that those two expressions form a Fourier transform pair. This is achieved by putting  $k = l$  in (7.17) so that

$$\begin{aligned} \phi(l) &= \frac{1}{T_0} \sum_{r=0}^{K-1} F(r) W^{-rl} \\ &= \frac{1}{T_0} \sum_{r=0}^{K-1} T \sum_{k=0}^{K-1} \phi(k) W^{r(k-l)} \\ &= \frac{1}{K} \sum_{k=0}^{K-1} \phi(k) \sum_{r=0}^{K-1} W^{r(k-l)} \end{aligned}$$

The second sum in this expression is zero for  $k \neq l$ ; when  $k = l$  it is  $K$ , so that the right hand side of the equality then becomes  $\phi(l)$  as required. An interesting aspect of this development has been that  $T$  has cancelled out, leaving  $1/K$  as the net constant from the forward and inverse transforms. As a result (7.15) and (7.17) could conveniently be written

$$F(r) = \sum_{k=0}^{K-1} \phi(k) W^{rk}, r = 0, \dots, K-1 \quad (7.15')$$

$$\phi(k) = \frac{1}{K} \sum_{r=0}^{K-1} F(r) W^{-rk}, k = 0, \dots, K-1 \quad (7.17')$$

### 7.7.3

#### Properties of the Discrete Fourier Transform

Three properties of the discrete Fourier transform and its inverse are of importance here.

**Linearity:** Both the DFT and DIFT are linear operations. Thus if  $F_1(r)$  is the DFT of  $\phi_1(k)$  and  $F_2(r)$  is the DFT of  $\phi_2(k)$  then for any complex constants  $a$  and  $b$ ,  $aF_1(r) + bF_2(r)$  is the DFT of  $a\phi_1(k) + b\phi_2(k)$ .

**Periodicity:** From (7.16),  $W^K = 1$  and  $W^{kK} = 1$  for  $k$  integral. Thus for  $r' = r + K$

$$F(r') = T \sum_{k=0}^{K-1} \phi(k) W^{(r+K)k} = F(r).$$

Therefore in general

$$F(r + mK) = F(r) \quad (7.18a)$$

$$\phi(k + mK) = \phi(k) \quad (7.18b)$$

where  $m$  is an integer. Thus both the sequence of time samples and the sequence of frequency samples are periodic with period  $K$ . This is consistent with the development of Sect. 7.7.1 and has two important implications. First, to generate the Fourier series components of a periodic function, samples need only be taken over one period. Secondly, sampling converts an aperiodic sequence into a periodic one, the period being determined by the sampling duration.

*Symmetry:* Let  $r' = K - r$  in (7.15), to give  $F(r') = T \sum_{k=0}^{K-1} \phi(k) W^{-rk} W^{kK}$ .

Since  $W^{kK} = 1$  this shows  $F(K - r) = F(r)^*$  where here  $*$  represents complex conjugate. This implies that the amplitude spectrum is symmetric about  $K/2$  and the phase spectrum is antisymmetric (i.e. odd).

#### 7.7.4 Computation of the Discrete Fourier Transform

It is convenient to consider the reduced form of (7.15):

$$A(r) = \frac{1}{T} F(r) = \sum_{k=0}^{K-1} \phi(k) W^{rk}, \quad r = 0, \dots, K-1 \quad (7.19)$$

Computation of the  $K$  values of  $A(r)$  from the  $K$  samples  $\phi(k)$  requires  $K^2$  multiplications and  $K^2$  additions, assuming that the required values of  $W^{rk}$  would have been calculated beforehand and stored. Since the  $W^{rk}$  are complex, the multiplications and additions necessary to evaluate  $A(r)$  are complex. Thus, as the number of samples  $\phi(k)$  becomes large, the time required to compute the sampled spectrum  $A(r)$  increases enormously (as the square of the number of samples). Between 1000 and 10,000 samples may in fact require unacceptably high computing time. A technique is required therefore to reduce substantially the number of arithmetic operations required in computing discrete Fourier transforms.

#### 7.7.5 Development of the Fast Fourier Transform Algorithm

Assume  $K$  is even; in fact the algorithm to follow will require  $K$  to be expressible as  $K = 2^m$  where  $m$  is an integer. From  $\phi(k)$  form two sequences  $Y(k)$  and  $Z(k)$  each of  $K/2$  samples. The first contains the even numbered samples of  $\phi(k)$  and the second the odd numbered samples, viz.

$$Y(k) : \phi(0), \phi(2), \dots, \phi(K-2)$$

$$Z(k) : \phi(1), \phi(3), \dots, \phi(K-1)$$

so that

$$\begin{aligned} Y(k) &= \phi(2k) \\ Z(k) &= \phi(2k+1) \end{aligned} \quad k = 0, \dots, \frac{K}{2} - 1.$$

Equation (7.19) can then be written

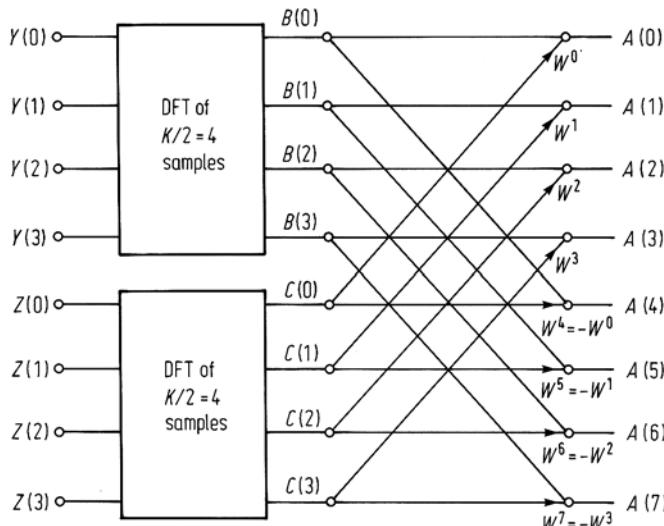
$$\begin{aligned} A(r) &= \sum_{k=0}^{K/2-1} \{Y(k) W^{2rk} + Z(k) W^{r(2k+1)}\} \\ &= \sum_{k=0}^{K/2-1} Y(k) W^{2rk} + W^r \sum_{k=0}^{K/2-1} Z(k) W^{2rk} \\ &= B(r) + W^r C(r) \end{aligned}$$

where  $B(r)$  and  $C(r)$  will be recognised as the discrete Fourier transforms of the sequences  $Y(k)$  and  $Z(k)$ . These are periodic, with period  $K/2$ , according to (7.18). Since  $W^{K/2} = -1$  it can be shown that the first  $K/2$  samples of  $A(r)$  and the last  $K/2$  samples of  $A(r)$  can be obtained from the same amount of computation, viz;

$$\left. \begin{aligned} A(r) &= B(r) + W^r C(r) \\ A\left(r + \frac{K}{2}\right) &= B(r) - W^r C(r) \end{aligned} \right] r = 0, \dots, \frac{K}{2} - 1 \quad (7.20)$$

Furthermore values of  $W^r$  only up to  $W^{K/2}$  are required.

The procedure of (7.20) can be represented conveniently in flow chart form. This is shown for  $K = 8$  in Fig. 7.8



**Fig. 7.8.** Flow chart for the first stage in the development of the fast Fourier transform algorithm, for the case of  $K = 8$

Equation (7.20) requires the Fourier transforms  $B(r)$  and  $C(r)$ . The same procedure can again be used to advantage for these;  $Y(k)$  and  $Z(k)$  are each broken up into sequences of odd and even samples, requiring  $Y(k)$  and  $Z(k)$  to contain an even number each. This in turn means that  $K$  had to be divisible at least by 4. Let  $S(k)$  contain the even numbered samples of  $Y(k)$  and  $T(k)$  the odd numbered samples. Also let  $U(k)$  contain the even numbered samples of  $Z(k)$  and  $V(k)$  the odd numbered samples:

$$\begin{aligned} S(k) &: Y(0), Y(2), \dots && (\text{i.e. } \phi(0), \phi(4), \dots) \\ T(k) &: Y(1), Y(3), \dots && (\text{i.e. } \phi(2), \phi(6), \dots) \\ U(k) &: Z(0), Z(2), \dots && (\text{i.e. } \phi(1), \phi(5), \dots) \\ V(k) &: Z(1), Z(3), \dots && (\text{i.e. } \phi(3), \phi(7), \dots) \end{aligned}$$

If the discrete Fourier transforms of these are denoted  $D(r)$ ,  $E(r)$ ,  $G(r)$  and  $H(r)$  respectively, each containing  $K/4$  points, then

$$\begin{aligned} B(r) &= \sum_{k=0}^{K/2-1} Y(k) W^{2rk} \\ &= D(r) + W^{2r} E(r) \end{aligned}$$

which can be written

$$\left. \begin{aligned} B(r) &= D(r) + W^{2r} E(r) \\ B\left(r + \frac{K}{4}\right) &= D(r) - W^{2r} E(r) \end{aligned} \right] r = 0, \dots, \frac{K}{4} - 1$$

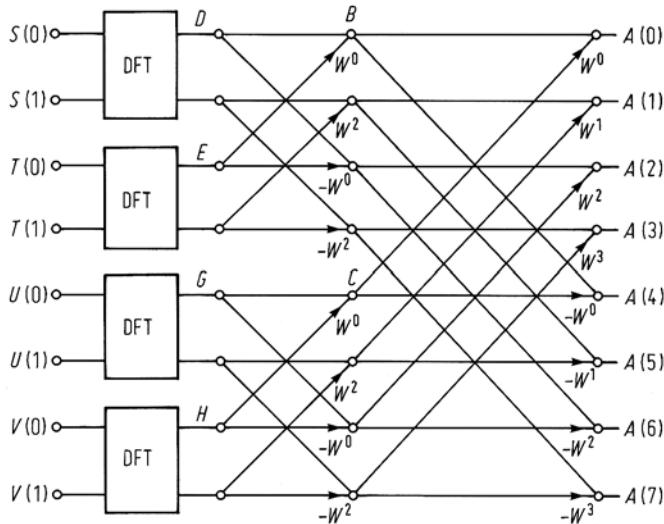
again showing that the first and second halves of the set of  $B(r)$  can be obtained by the same calculations. Similarly

$$\left. \begin{aligned} C(r) &= G(r) + W^{2r} H(r) \\ C\left(r + \frac{K}{4}\right) &= G(r) - W^{2r} H(r) \end{aligned} \right] r = 0, \dots, \frac{K}{4} - 1.$$

Figure 7.9 shows how the flow chart of Fig. 7.8 can be modified to take account of this development.

Clearly the procedure followed to this point can be repeated as many times as there are discrete Fourier transforms left to compute. Ultimately transforms will be required on sequences with just two samples each. For example if  $K = 8$ , the sequences  $S$ ,  $T$ ,  $U$  and  $V$  will each contain only two samples and their discrete Fourier transforms will be of the form

$$\begin{aligned} D(r) &= \sum_{k=0}^1 S(k) W^{4rk}, & r = 0, 1 \\ &= S(0) W^0 + S(1) W^{4r} & r = 0, 1 \end{aligned}$$

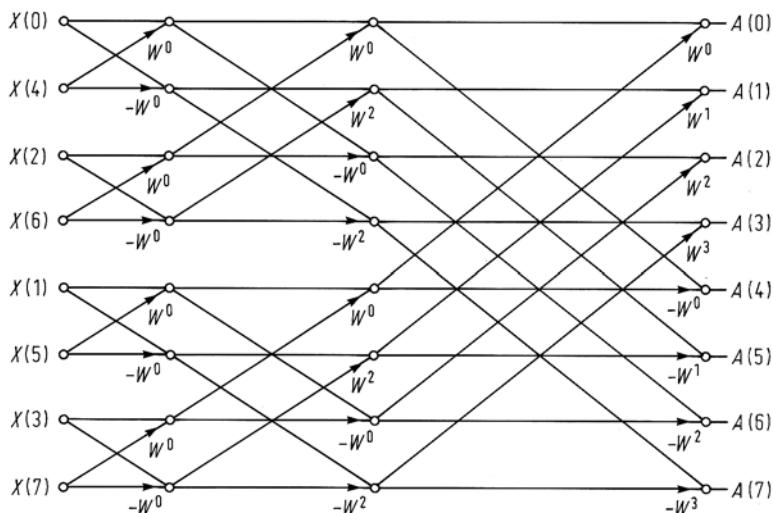


**Fig. 7.9.** Flow chart for the second stage of the development of the fast Fourier transform algorithm, for the case of  $K = 8$

$$\text{i.e. } D(0) = S(0) + S(1)$$

$$D(1) = S(0) - S(1),$$

showing that the discrete Fourier transform of two samples is obtained by simple addition and subtraction. Doing likewise for the other sequences gives the final flow chart for  $K = 8$  as shown in Fig. 7.10.



**Fig. 7.10.** Flow chart for a complete fast Fourier transform evaluation when  $K = 8$

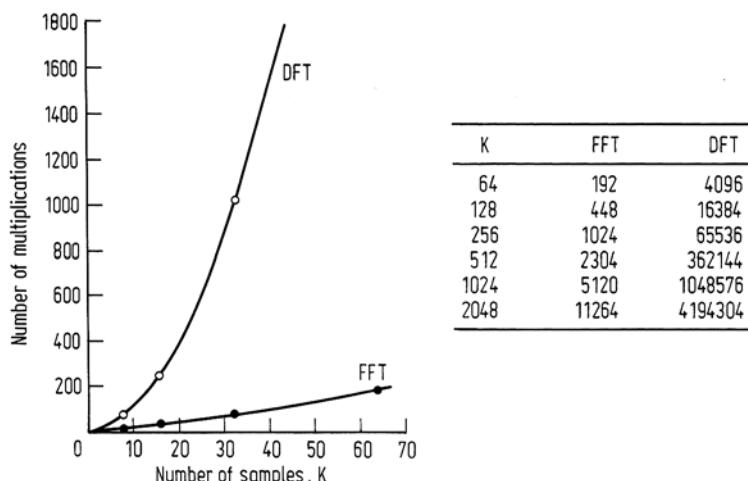
### 7.7.6

#### Computational Cost of the Fast Fourier Transform

The information contained in Fig. 7.10 can be used to determine the computational cost of the fast Fourier transform algorithm and therefore to find its speed advantage over a direct evaluation of the discrete formula in (7.19). The figure shows that the only multiplications required are by the values of  $W$ . While this is strictly not necessary in the first set of calculations on the left of the figure (Since  $W^0 = 1$ , and  $-W^0 = -1$ ) it is simpler in programming if the multiplications are retained. Thus the left hand set of computations requires  $K/2$  complex multiplications and  $K$  additions (or subtractions). The next column of operations requires another  $K/2$  multiplications as does the last set for the case of  $K = 8$ . Altogether for this illustration  $3/2K$  multiplications and  $3K$  additions are required. It is easy to generalize this to:

$$\begin{aligned} \text{number of complex multiplications} &= \frac{1}{2} K \log_2 K \\ \text{number of complex additions} &= K \log_2 K. \end{aligned}$$

On the basis of multiplications alone the fast Fourier transform (FFT) is seen, from the material in Sect. 7.7.4, to be faster than direct evaluation of the discrete Fourier transform (DFT) by a factor of  $2K/\log_2 K$ . Moreover its cost increases almost linearly with the number of samples, whereas that for the DFT increases quadratically. This is illustrated in Fig. 7.11.



**Fig. 7.11.** Number of multiplications required in the evaluation of a discrete Fourier transform directly (DFT) and by means of the fast Fourier transform method (FFT)

### 7.7.7

#### Bit Shuffling and Storage Considerations

Application of the fast Fourier transform requires  $K$  to be continuously divisible by 2 (i.e.  $K = 2^m$  as indicated above). Although other versions of the algorithm can also be derived (Brigham, 1974) the case of  $K = 2^m$  is most common, and is used here.

Inspection of the flow chart in Fig. 7.10 reveals that the order of the data fed into the algorithm needs to be rearranged before the technique can be employed. This can be achieved very simply by a process known as bit shuffling. To do this the index of the input samples is expressed in binary notation (see Appendix C), the binary digits are reversed, and the new binary number converted back to decimal form, as illustrated in the following for  $K = 8$ .

$$\begin{array}{llll}
 X(0) & \rightarrow X(000) & \rightarrow X(000) & \rightarrow X(0) \\
 X(1) & X(001) & X(100) & X(4) \\
 X(2) & X(010) & X(010) & X(2) \\
 X(3) & X(011) & X(110) & X(6) \\
 X(4) & X(100) & X(001) & X(1) \\
 X(5) & X(101) & X(101) & X(5) \\
 X(6) & X(110) & X(011) & X(3) \\
 X(7) & X(111) & X(111) & X(7)
 \end{array}$$

Apart from the immense savings in time, use of the FFT also leads to a savings in memory. Apart from storing the  $K/2$  values of  $W^r$  the entire computation can be carried out using a complex vector of length  $K + 1$ . This is because there exist pairs of elements in each vector or column of the operation whose values are computed from numbers stored in the same pair of locations in the previous column.

## 7.8

### The Discrete Fourier Transform of an Image

#### 7.8.1

##### Definition

The previous sections have treated functions with a single independent variable. That variable could have been time, or even position along a line of an image. We now need to turn our attention to functions with two independent variables, to allow Fourier transforms of images to be determined. Despite this apparent increase in complexity we will find that full advantage can be taken of the material of the previous sections. Let

$$\phi(i, j), \quad i, j = 0, \dots, K - 1 \tag{7.21}$$

be the brightness of a pixel at location  $i, j$  in an image of  $K \times K$  pixels. The Fourier

transform of the image, in discrete form, is described by

$$\Phi(r, s) = \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \phi(i, j) \exp [-j2\pi(ir + js)/K]. \quad (7.22)$$

An image can be reconstructed from its transform according to

$$\phi(i, j) = \frac{1}{K^2} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} \Phi(r, s) \exp [+j2\pi(ir + js)/K]. \quad (7.23)$$

### 7.8.2

#### Evaluation of the Two Dimensional, Discrete Fourier Transform

Equation (7.22) can be rewritten as

$$\Phi(r, s) = \sum_{i=0}^{K-1} W^{ir} \sum_{j=0}^{K-1} \phi(i, j) W^{js} \quad (7.24)$$

with  $W = e^{-j2\pi/K}$  as before. The term involving the right hand sum can be recognised as the one dimensional discrete Fourier transform

$$\Phi(i, s) = \sum_{j=0}^{K-1} \phi(i, j) W^{js}, \quad i = 0, \dots, K-1. \quad (7.25)$$

In fact it is the one dimensional transform of the  $i$ th row of pixels in the image. The result of this operation is that the rows of an image are replaced by their Fourier transforms; the transformed pixels are then addressed by the spatial frequency index  $s$  across a row rather than by the positional index  $j$ . Using (7.25) in (7.24) gives

$$\Phi(r, s) = \sum_{i=0}^{K-1} \Phi(i, s) W^{ir} \quad (7.26)$$

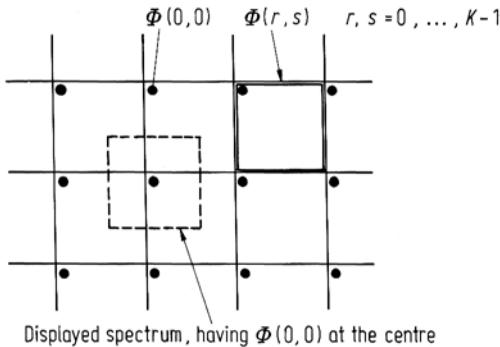
which is the one dimensional discrete Fourier transform of the  $s$ th column of the image, after the row transforms of (7.25) have been performed.

Thus, to compute the two dimensional Fourier transform of an image, it is only necessary to transform each row individually to generate an intermediate image, and then transform this by column to yield the final result. Both the row and column transforms would be carried out using the fast Fourier transform algorithm of Sect. 7.7.5. From the information provided in Sect. 7.7.6 it can be seen therefore that the number of multiplications required to transform an image is  $K^2 \log_2 K$ .

### 7.8.3

#### The Concept of Spatial Frequency

Entries in the Fourier transformed image  $\Phi(r, s)$  represent the composition of the original image in terms of spatial frequency components, both vertically and horizontally. Spatial frequency is the image analog of the frequency of a signal in time.

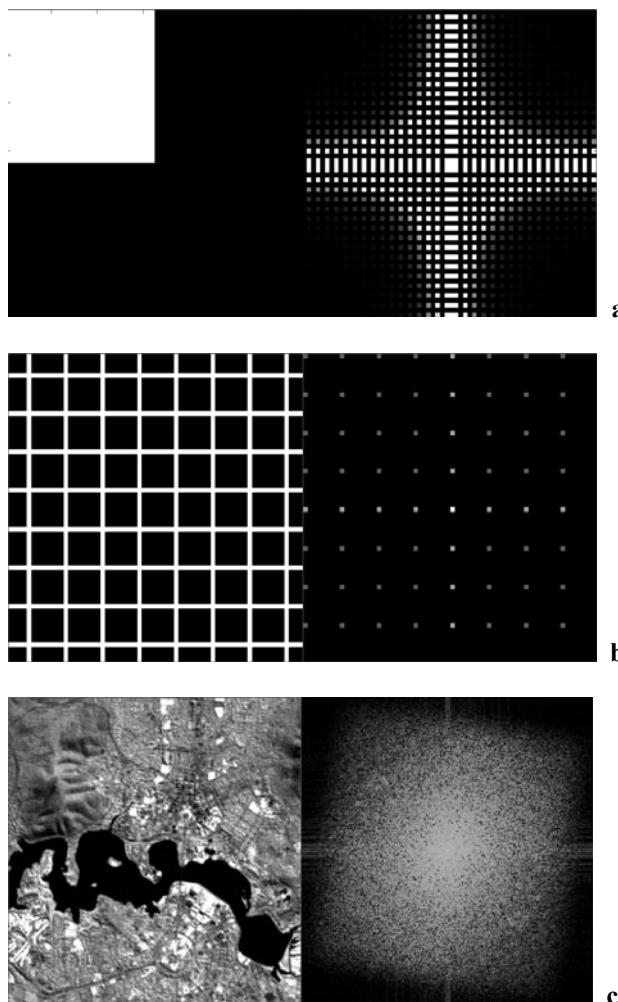


**Fig. 7.12.** Illustration of the periodic nature of the two dimensional discrete Fourier transform, showing how an array centred on  $\Phi(0,0)$  is chosen for symmetrical display purposes

A sinusoidal signal with high frequency alternates rapidly, whereas a low frequency signal changes slowly with time. Similarly, an image with high spatial frequency, say in the horizontal direction, exhibits frequent changes of brightness with position horizontally. A picture of a crowd of people would be a particular example. By comparison a head and shoulders view of a person is likely to be characterised mainly by low spatial frequencies. Typically an image is composed of a collection of both horizontal and vertical spatial frequency components of differing strengths and these are what the discrete Fourier transform indicates. The upper left hand pixel in  $\Phi(r, s)$  – i.e.  $\Phi(0, 0)$  – is the average brightness value of the image. This is the component in the spectrum with zero frequency in both directions. Thereafter pixels of  $\Phi(r, s)$  both horizontally and vertically represent components with frequencies that increment by  $1/K$  where the original image is of size  $K \times K$ . Should the scale of the image be known then the spatial frequency increment can be calibrated in terms of  $\text{metres}^{-1}$ . For example the increment in spatial frequency for a  $512 \times 512$  pixel image that covers 15.36 km (i.e. Landsat TM) is  $65 \times 10^{-6} \text{ m}^{-1}$ .

In Sect. 7.7.3 it is shown that the one dimensional discrete Fourier transform is periodic with period  $K$ . The same is true of the discrete two dimensional form. Indeed the  $K \times K$  pixels of  $\Phi(r, s)$  computed according to (7.22) can be viewed as one period of an infinite periodic two dimensional array in the manner depicted in Fig. 7.12. It is also shown that the amplitude of the discrete Fourier transform is symmetric about  $K/2$ . Similarly  $\Phi(r, s)$  is symmetric about its centre. This can be interpreted by implying that no new amplitude information is shown by displaying pixels horizontally and vertically beyond  $K/2$ . Rather than ignore them (since their accompanying phase is important) the display is adjusted in the manner shown in Fig. 7.12 to bring  $\Phi(0, 0)$  to the centre. In this way the pixel at the centre of the Fourier transform array represents the image average brightness value. Pixels away from the centre represent the proportions of increasing spatial frequency components in the image. This is the usual method of presenting two dimensional image transforms. Examples of spectra displayed in this manner are given in Fig. 7.13. To make visible components with smaller amplitudes, a logarithmic amplitude scaling has been used, according to (Gonzalez and Woods, 1992)

$$D(r, s) = \log [1 + |\Phi(r, s)|].$$



**Fig. 7.13.** Illustrations of Fourier transforms of images. **a** Single square; **b** Bar pattern; **c** Hymap image

#### 7.8.4

#### Image Filtering for Geometric Enhancement

The high spatial frequency content of an image is that associated with frequent changes of brightness with position. Edges, lines and some types of noise are examples of high frequency data. In contrast, gradual changes of brightness with position, such as associated with more general tonal variations, account for the low frequency content in the spectrum. Since ranges of spatial frequency are identified with regions in the spectrum we can envisage how the spectrum of an image could be altered to produce different geometric enhancements of the image itself. For example, if the

region near the centre of the spectrum is removed, leaving behind only the high frequencies, and the image is then reconstructed from the modified spectrum, a version containing only edges and line-like features will be produced. On the other hand, if the high frequency components are removed, leaving behind only the region near the centre of the spectrum, the reconstructed image will appear smoothed, since edges, lines and high frequency noise will have been deleted.

Modification of the spectrum in the manner just described can be expressed as a multiplicative operation:

$$Y(r, s) = \Phi(r, s) H(r, s) \quad \text{for all } r, s \quad (7.27)$$

where  $H(r, s)$  is the filter function and  $Y(r, s)$  is the new spectrum. To implement simple sharpening or smoothing as described above  $H(r, s)$  would be set to 0 for those frequency components to be removed and 1 for those components to be retained. Often sharpening is called high pass filtering, and smoothing low pass filtering, because of the nature of the modification to the spectrum. Both can be implemented also with the template methods of Chap. 5. However (7.27) allows more complicated filtering operations to be carried out. As an example, a specific band of spatial frequency could be excluded readily.  $H(r, s)$  can also be chosen to have values other than 0 and 1 to allow more versatile modification of the spectrum.

The overall process of geometric enhancement via the frequency domain involves three steps. First, the image has to be Fourier transformed to produce its spectrum. Secondly, the spectrum is modified according to (7.27). Finally the image is reconstructed from the modified spectrum using (7.23), which can also be implemented by rows and columns. Together these three operations require  $2K^2 \log_2 K + K^2$  multiplications, as used in Sect. 5.4 to compare this approach to that based upon simple templates.

### 7.8.5

#### Convolution in Two Dimensions

The convolution theorem for functions (Sect. 7.5.3) has a two dimensional counterpart, again in two forms. These are:

If

$$y(i, j) = \phi(i, j) * h(i, j)$$

then

$$Y(r, s) = \Phi(r, s) H(r, s) \quad (7.28a)$$

and, if

$$y(i, j) = \phi(i, j) h(i, j)$$

then

$$Y(r, s) = \Phi(r, s) * H(r, s). \quad (7.28b)$$

Unlike (7.10b) there is no  $1/2\pi$  scaling factor here since the spatial frequency variables  $r$  and  $s$  are equivalent to frequency  $f$  in Hz and not the radian frequency  $\omega$  in  $\text{rad} \cdot s^{-1}$  used in (7.10b).

The convolution operation implied in (7.28) is defined by (5.3). However when digital images are of concern its discrete version is of interest. This is defined, in the image domain, as

$$y(i, j) = \sum_m \sum_n \phi(m, n) h(i - m, j - n) \quad (7.29)$$

where  $m$  and  $n$  are dummy variables. As with one dimensional convolution described in Sect. 7.5.1 evaluation of (7.29) requires that one function, in this case the filter function, be folded about the origin (which in two dimensions amounts to a  $180^\circ$  rotation) to produce  $h(-m, -n)$  and then delayed by variable amounts  $i, j$ . The delayed folded version is then multiplied pixel by pixel with the image  $\phi(m, n)$  and the sum over all spectral pixels taken. This produces one pixel  $y(i, j)$  in the modified image.

Equation (7.28) implies that any of the geometric enhancement operations that can be carried out by modifying the spectrum can also be carried out by performing a convolution between the image and the inverse Fourier transform of the filter function  $H(r, s)$ . Conversely, operations such as simple mean value filtering with an  $M \times N$  template as described in Sect. 5.5.1, can also be described in the spatial frequency domain. This requires the Fourier transform of the template to be found. To do this requires the template to be regarded as of the same dimensions as the image but with a value of zero everywhere except for a set of  $M \times N$  pixels with the appropriate non-zero value.

## 7.9 Concluding Remarks

Geometric modification of an image via the frequency domain is a particularly powerful technique owing to the ease with which the filter function  $H(r, s)$  may be designed. The material presented in this Chapter has been intended as an introduction to the concepts and operations involved. For the user contemplating using Fourier domain methods, several other issues should be taken into consideration including the use of so-called window functions. This is illustrated most easily by a return to the material on sampling in Sect. 7.6. In that section it was noted that a sampled function could be regarded as the unsampled version multiplied by an infinite periodic sequence of impulses. The spectrum of the infinite set of samples so produced is the spectrum of the original function convolved with the spectrum of the sequence of impulses as shown in Fig. 7.6. However in practice it is not possible to take an infinite number of samples of a function. Instead sampling is commenced at a given time and terminated after some period  $\tau$ . This finite time sampling window can be considered as a long pulse of unit amplitude and duration  $\tau$  that multiplies the infinite sequence of samples. The spectrum of the set of samples is, as a consequence, modified by being convolved by the spectrum of the sampling window. Since the window is a

long pulse, its Fourier transform is as shown in Fig. 7.4 although compressed to near the origin. If the sampling duration is long enough this approximates an impulse and there is little effect on the spectrum. For shorter sampling times however the side-lobe in Fig. 7.4b cause distortion of the spectrum. To minimise this effect sampling windows different to a long pulse are sometimes used. A good consideration of these is found in Brigham (1974).

In the preceding sections we have referred to the Fourier transform approach as a means for geometric enhancement since it can implement operations such as sharpening and smoothing. In the material of Chapter Five these are referred to explicitly as neighbourhood operations. To appreciate that the Fourier transform is also a neighbourhood operation consider the flow chart for the fast Fourier transform implementation in Fig. 7.10. If we pick one output value – i.e. one point on the spectrum – it can be traced back through the flow chart and be seen to have a contribution from every one of the input samples. In a similar manner the pixels in the Fourier transform of an image have contributions from all of the pixels in the original image.

Other transformations also exist, perhaps the most notable in the past few years being the wavelet transform. The theory of the wavelet transformation can be quite detailed, especially if generalised beyond the field of real functions. However, several excellent treatments are available when the transform is to be applied to real image data, perhaps the most accessible of which is that given by Castleman (1996).

The wavelet transform is important in so far that it provides a compact description of signals (or images) that are limited in time (or spatial extent). The following introduces the concept; Castleman should be consulted for details, including how the transform is defined, and how it can be used and computed in practice. It finds application in image compression and coding, and in the detection of localised image features.

Suppose you listen to an organ playing a single, pure tone. As a function of time it will be sinusoidal for as long as the key is pressed. We could, if we wished, envisage that the sinusoid started at minus infinity and goes to plus infinity in time. It is the simplest of all signals in terms of Fourier analysis and its Fourier series is a single frequency (with positive and negative frequency components) as an application of (7.7b) will show.

Now suppose you hear a piano play a single note. Rather than lasting for all time, the time waveform of the piano note would be a time limited sinusoid. We can still find its Fourier components – i.e. the set of frequencies it is composed of, by noting that it is the product of an infinitely long sinusoid and the unit pulse waveform of Fig. 7.4a. Application of (7.10b) and the material of Sect. 7.5.2 shows that its spectrum will be the function of Fig. 7.4b but with its “origin” shifted to the frequency of the sinusoid. The spectrum of the time limited signal is now unlimited, although it does drop off quickly as frequency goes to plus and minus infinity.

To represent short time signals, like the piano note, by a Fourier series or transform, although theoretically acceptable, is cumbersome. Yet that is a problem because many signals (such as speech) consists of limited time signals, just as images are also limited spatially.

The invention of the wavelet – i.e. a small wavelike signal that is limited in time, has a frequency-like property and is defined to occur at a certain time – is meant to make the description of such signals easier and to assist in the identification of signal characteristics that occur at particular times. In the case of images, wavelets help in the description of properties that are highly localised such as edges and lines.

## References for Chapter 7

Treatments of digital image processing in the fields of electrical engineering and computer science invariably contain detailed considerations of the use of the Fourier transform and frequency domain techniques for geometric modification of image data. Particular texts that could be consulted include Castleman (1996), Gonzalez and Woods (1992) and Moik (1980). An excellent presentation of the discrete Fourier transform, discrete convolution and the fast Fourier transform algorithm will be found in Brigham (1974, 1988). While Brigham relates to the one dimensional case it will be clear from the material in Sect. 7.8.2 above that it can be used also with images.

- E.O. Brigham, 1974: The Fast Fourier Transform. N.J. Prentice-Hall.
- E.O. Brigham, 1988: The Fast Fourier Transform and its Applications. N.J. Prentice-Hall.
- K.R. Castleman, 1996: Digital Image Processing. N.J. Prentice-Hall.
- R.C. Gonzalez and R.E. Woods, 1992: Digital Image Processing. Mass. Addison-Wesley.
- C.D. McGillem and G.R. Cooper, 1984: Continuous and Discrete Signal and System Analysis. N.Y. Holt, Rinehart and Winston.
- J.G. Moik, 1980: Digital Processing of Remotely Sensed Images. Washington, NASA.
- A. Papoulis, 1980: Circuits and Systems: A Modern Approach. Tokyo, Holt-Saunders.

## Problems

**7.1** Compute the discrete Fourier transform of the square wave shown in Fig. 7.3 using  $K = 2, 4$  and  $8$  samples per period of the waveform respectively. You can use the flow chart of Fig. 7.10 to help in this.

**7.2** Compute the discrete Fourier transform of the unit pulse shown in Fig. 7.4. Use respectively  $K = 2, 4$  and  $8$  samples over a time interval equal to  $8a$ , where  $2a$  is the width of the pulse as shown in the Figure. Compare the results with those obtained in problem 7.1.

**7.3** (a) A common technique for smoothing an image is to compute averages over square or rectangular windows as discussed in Sect. 5.5. Consider a  $3 \times 1$  smoothing template used to smooth a single line of image data in the manner of Fig. 5.4. Determine the corresponding filter function in the spatial frequency domain by finding the discrete Fourier transform of the template. You may find the material of Fig. 7.4 to be of value.

(b) Imagine an ideal low pass filter function in the spatial frequency domain that could be used to smooth just the lines of an image. Determine the corresponding function in the image domain by computing the inverse Fourier transform of the ideal filter. Taking into account the discrete pixel nature of the image, approximate the inverse transform by an appropriate one dimensional template.

**7.4** Verify the results in Sect. 7.5.2 graphically.

**7.5** (a) The periodic sequence of impulses of (7.11) is an idealised sampling function. In practice it is not possible to take infinitessimally short samples of a function; rather the samples will have a finite, albeit small duration. This could be modelled mathematically by replacing  $\Delta(t)$  in (7.12) by a periodic pulse waveform. This periodic sequence of pulses can be represented by the convolution of a single pulse with the periodic sequence of impulses in (7.11). With this in mind describe what modifications are needed to Fig. 7.6 to account for samples of finite duration.

(b) Suppose the total period of sampling is equivalent to ten sample intervals. Describe the effect this has on Fig. 7.6.

**7.6** In Fig. 7.6a suppose the function  $f(t)$  is a sinewave of frequency  $B$  Hz. Its frequency spectrum will consist of two impulses, one at  $+B$  Hz and the other at  $-B$  Hz. Produce the spectrum of the sampled sinusoid if only three samples are taken every two periods. Suppose the waveform is then reconstructed by feeding the samples through a low pass filter that will pass all frequency components unattenuated, up to  $1/2T$  Hz, where  $T$  is the sampling interval, and will exclude all components with frequencies in excess of  $1/2T$  Hz. Describe the shape of the reconstructed signal; this will give an appreciation of aliasing distortion.

# 8

## Supervised Classification Techniques

The purpose of this chapter is to present the algorithms used for the supervised classification of single sensor remote sensing image data.

When data from a variety of sensors or sources (such as found in the integrated spatial data base of a Geographical Information System) requires analysis, more sophisticated tools may be required. These are the subject of Chap. 12 which deals with the topic of Multisource Classification.

### 8.1

#### Steps in Supervised Classification

Supervised classification is the procedure most often used for quantitative analysis of remote sensing image data. It rests upon using suitable algorithms to label the pixels in an image as representing particular ground cover types, or classes. A variety of algorithms is available for this, ranging from those based upon probability distribution models for the classes of interest to those in which the multispectral space is partitioned into class-specific regions using optimally located surfaces. Irrespective of the particular method chosen, the essential practical steps usually include:

1. Decide the set of ground cover types into which the image is to be segmented. These are the information classes and could, for example, be water, urban regions, croplands, rangelands, etc.
2. Choose representative or prototype pixels from each of the desired set of classes. These pixels are said to form *training data*. Training sets for each class can be established using site visits, maps, air photographs or even photointerpretation of a colour composite product formed from the image data. Often the training pixels for a given class will lie in a common region enclosed by a border. That region is then often called a *training field*.
3. Use the training data to estimate the parameters of the particular classifier algorithm to be used; these parameters will be the properties of the probability model

- used or will be equations that define partitions in the multispectral space. The set of parameters for a given class is sometimes called the *signature* of that class.
4. Using the trained classifier, label or classify every pixel in the image into one of the desired ground cover types (information classes). Here the whole image segment of interest is typically classified. Whereas training in Step 2 may have required the user to identify perhaps 1% of the image pixels by other means, the computer will label the rest by classification.
  5. Produce tabular summaries or thematic (class) maps which summarise the results of the classification.
  6. Assess the accuracy of the final product using a labelled testing data set.

In practice it might be necessary to decide, on the basis of the results obtained at Step 6, to refine the training process in order to improve classification accuracy. Sometimes it might even be desirable to classify just the training samples themselves to ensure that the signatures generated at Step 3 are adequate.

It is our objective now to consider the range of algorithms that could be used in 3 and 4. In so doing it will be assumed that the information classes each consists of only one spectral class, so that the two names will be used synonomously. (See Chap. 3 for a discussion of the two class types.) By making this assumption, problems with establishing sub-classes will not distract from the algorithm development to be given. Handling sub-classes is taken care of explicitly in Chaps. 9 and 11.

In the following sections it is assumed that the reader is familiar at least with the sections on quantitative analysis contained in Chap. 3. This relates particularly to definitions and terminology.

## 8.2 Maximum Likelihood Classification

Maximum likelihood classification is the most common supervised classification method used with remote sensing image data. This is developed in the following in a statistically acceptable manner; it can be derived however in a more general and rigorous manner and this is presented for completeness in Appendix F. The present approach is sufficient though for most remote sensing exercises.

### 8.2.1 Bayes' Classification

Let the spectral classes for an image be represented by

$$\omega_i, i = 1, \dots, M$$

where  $M$  is the total number of classes. In trying to determine the class or category to which a pixel vector  $\mathbf{x}$  belongs it is strictly the conditional probabilities

$$p(\omega_i | \mathbf{x}), i = 1, \dots, M$$

that are of interest. The measurement vector  $\mathbf{x}$  is a column of brightness values for the pixel. It describes the pixel as a point in multispectral space with co-ordinates defined by the brightnesses, as shown in the simple two-dimensional example of Fig. 3.5. The probability  $p(\omega_i|\mathbf{x})$  gives the likelihood that the correct class is  $\omega_i$  for a pixel at position  $\mathbf{x}$ . Classification is performed according to

$$\mathbf{x} \in \omega_i, \quad \text{if } p(\omega_i|\mathbf{x}) > p(\omega_j|\mathbf{x}) \quad \text{for all } j \neq i \quad (8.1)$$

i.e., the pixel at  $\mathbf{x}$  belongs to class  $\omega_i$  if  $p(\omega_i|\mathbf{x})$  is the largest. This intuitive *decision rule* is a special case of a more general rule in which the decisions can be biased according to different degrees of significance being attached to different incorrect classifications. The general approach is called Bayes' classification and is the subject of the treatment in Appendix F.

### 8.2.2

#### The Maximum Likelihood Decision Rule

Despite its simplicity, the  $p(\omega_i|\mathbf{x})$  in (8.1) are unknown. Suppose however that sufficient training data is available for each ground cover type. This can be used to estimate a probability distribution for a cover type that describes the chance of finding a pixel from class  $\omega_i$ , say, at the position  $\mathbf{x}$ . Later the form of this distribution function will be made more specific. For the moment however it will be retained in general terms and represented by the symbol  $p(\mathbf{x}|\omega_i)$ . There will be as many  $p(\mathbf{x}|\omega_i)$  as there are ground cover classes. In other words, for a pixel at a position  $\mathbf{x}$  in multispectral space a set of probabilities can be computed that give the relative likelihoods that the pixel belongs to each available class.

The desired  $p(\omega_i|\mathbf{x})$  in (8.1) and the available  $p(\mathbf{x}|\omega_i)$  - estimated from training data – are related by Bayes' theorem (Freund, 1992):

$$p(\omega_i|\mathbf{x}) = p(\mathbf{x}|\omega_i) p(\omega_i)/p(\mathbf{x}) \quad (8.2)$$

where  $p(\omega_i)$  is the probability that class  $\omega_i$  occurs in the image. If, for example, 15% of the pixels of an image happen to belong to class  $\omega_i$  then  $p(\omega_i) = 0.15$ ;  $p(\mathbf{x})$  in (8.2) is the probability of finding a pixel from *any* class at location  $\mathbf{x}$ . It is of interest to note in passing that

$$p(\mathbf{x}) = \sum_{i=1}^M p(\mathbf{x}|\omega_i) p(\omega_i),$$

although  $p(\mathbf{x})$  itself is not important in the following. The  $p(\omega_i)$  are called *a priori* or prior probabilities, since they are the probabilities with which class membership of a pixel could be guessed before classification. By comparison the  $p(\omega_i|\mathbf{x})$  are posterior probabilities. Using (8.2) it can be seen that the classification rule of (8.1) is:

$$\mathbf{x} \in \omega_i \quad \text{if } p(\mathbf{x}|\omega_i) p(\omega_i) > p(\mathbf{x}|\omega_j) p(\omega_j) \quad \text{for all } j \neq i \quad (8.3)$$

where  $p(\mathbf{x})$  has been removed as a common factor. The rule of (8.3) is more acceptable than that of (8.1) since the  $p(\mathbf{x}|\omega_i)$  are known from training data, and it

is conceivable that the  $p(\omega_i)$  are also known or can be estimated from the analyst's knowledge of the image. Mathematical convenience results if in (8.3) the definition

$$\begin{aligned} g_i(\mathbf{x}) &= \ln \{p(\mathbf{x}|\omega_i) p(\omega_i)\} \\ &= \ln p(\mathbf{x}|\omega_i) + \ln p(\omega_i) \end{aligned} \quad (8.4)$$

is used, where  $\ln$  is the natural logarithm, so that (8.3) is restated as

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i \quad (8.5)$$

This is, with one modification to follow, the decision rule used in maximum likelihood classification; the  $g_i(\mathbf{x})$  are referred to as *discriminant functions*.

### 8.2.3

#### Multivariate Normal Class Models

At this stage it is assumed that the probability distributions for the classes are of the form of multivariate normal models. This is an assumption, rather than a demonstrable property of natural spectral or information classes; however it leads to mathematical simplifications in the following. Moreover it is one distribution for which properties of the multivariate form are well-known.

In (8.4) therefore, it is now assumed for  $N$  bands that (see Appendix E)

$$p(\mathbf{x}|\omega_i) = (2\pi)^{-N/2} |\Sigma_i|^{-1/2} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) \right\} \quad (8.6)$$

where  $\mathbf{m}_i$  and  $\Sigma_i$  are the mean vector and covariance matrix of the data in class  $\omega_i$ . The resulting term  $-N/2 \ln (2\pi)$  is common to all  $g_i(\mathbf{x})$  and does not aid discrimination. Consequently it is ignored and the final form of the discriminant function for maximum likelihood classification, based upon the assumption of normal statistics, is:

$$g_i(\mathbf{x}) = \ln p(\omega_i) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) \quad (8.7)$$

Often the analyst has no useful information about the  $p(\omega_i)$ , in which case a situation of equal prior probabilities is assumed; as a result  $\ln p(\omega_i)$  can be removed from (8.7) since it is then the same for all  $i$ . In that case the 1/2 common factor can also be removed leaving, as the discriminant function:

$$g_i(\mathbf{x}) = -\ln |\Sigma_i| - (\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) \quad (8.8)$$

Implementation of the maximum likelihood decision rule involves using either (8.7) or (8.8) in (8.5). There is a further consideration however concerned with whether any of the available labels or classes is appropriate. This relates to the use of thresholds as discussed in Sect. 8.2.5 following.

### 8.2.4

#### Decision Surfaces

As a means for assessing the capabilities of the maximum likelihood decision rule it is of value to determine the essential shapes of the surfaces that separate one class

from another in the multispectral domain. These surfaces, albeit implicit, can be devised in the following manner.

Spectral classes are defined by those regions in multispectral space where their discriminant functions are the largest. Clearly these regions are separated by surfaces where the discriminant functions for adjoining spectral classes are equal. The  $i$ th and  $j$ th spectral classes are separated therefore by the surface

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0.$$

This is referred to as a *decision surface* since, if all the surfaces separating spectral classes are known, decisions about the class membership of a pixel can be made on the basis of its position relative to the complete set of surfaces.

The construction  $(\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i)$  in (8.7) and (8.8) is a quadratic function of  $\mathbf{x}$ . Consequently the decision surfaces implemented by maximum likelihood classification are quadratic and thus take the form of parabolas, circles and ellipses. Some indication of this can be seen in Fig. 3.8.

### 8.2.5 Thresholds

It is implicit in the foregoing development that pixels at every point in multispectral space will be classified into one of the available classes  $\omega_i$ , irrespective of how small the actual probabilities of class membership are. This is illustrated for one dimensional data in Fig. 8.1a. Poor classification can result as indicated. Such situations can arise if spectral classes (between 1 and 2 or beyond 3) have been overlooked or, if knowing other classes existed, enough training data was not available to estimate the parameters of their distributions with any degree of accuracy (see Sect. 8.2.6 following). In situations such as these it is sensible to apply thresholds to the decision process in the manner depicted in Fig. 8.1b. Pixels which have probabilities for *all* classes below the threshold are not classified.

In practice, thresholds are applied to the discriminant functions and not the probability distributions, since the latter are never actually computed. With the incorporation of a threshold therefore, the decision rule of (8.5) becomes

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all } j \neq i \quad (8.9a)$$

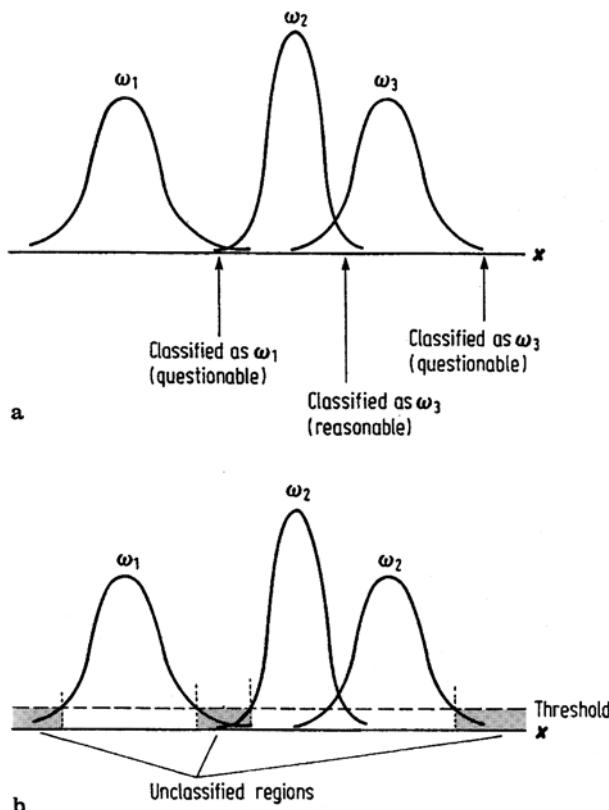
$$\text{and} \quad g_i(\mathbf{x}) > T_i \quad (8.9b)$$

where  $T_i$  is the threshold seen to be significant for spectral class  $\omega_i$ . It is now necessary to consider how  $T_i$  can be estimated. From (8.7) and (8.9b) a classification is acceptable if

$$\ln p(\omega_i) - \frac{1}{2} \ln |\Sigma_i| - \frac{1}{2} (\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) > T_i$$

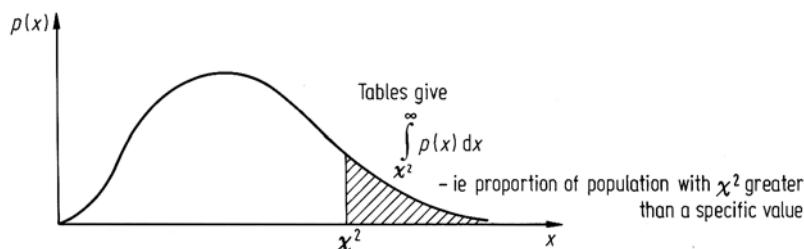
i.e.

$$(\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) < -2T_i - \ln |\Sigma_i| + 2 \ln p(\omega_i) \quad (8.10)$$



**Fig. 8.1.** a Illustration of poor classification for patterns lying near the tails of the distribution functions of all spectral classes; b Use of a threshold to remove poor classification

The left hand side of (8.10) has a  $\chi^2$  distribution with  $N$  degrees of freedom, if  $x$  is (assumed to be) distributed normally (Swain and Davis, 1978).  $N$  is the dimensionality of the multispectral space. As a result  $\chi^2$  tables can be consulted to determine that value of  $(x - \mathbf{m}_i)^t \Sigma_i^{-1} (x - \mathbf{m}_i)$  below which a desired percentage of pixels will exist (noting that larger values of that quadratic form correspond to pixels lying further out in the tails of the normal probability distribution). This is depicted in Fig. 8.2.



**Fig. 8.2.** Use of the  $\chi^2$  distribution for obtaining classifier thresholds

As an example of how this is used consider the need to choose a threshold such that 95% of all pixels in a class will be classified (i.e. such that the 5% least likely pixels for each spectral class will be rejected).  $\chi^2$  tables show that 95% of all pixels have  $\chi^2$  values (in Fig. 8.2) less than 9.488. Thus, from (8.10)

$$T_i = -4.744 - \frac{1}{2} \ln |\Sigma_i| + \ln p(\omega_i)$$

which thus can be calculated from a knowledge of the prior probability and covariance matrix of the  $i$ th spectral class.

### 8.2.6

#### Number of Training Pixels Required for Each Class

Sufficient training pixels for each spectral class must be available to allow reasonable estimates to be obtained of the elements of the class conditional mean vector and covariance matrix. For an  $N$  dimensional multispectral space the covariance matrix is symmetric of size  $N \times N$ . It has, therefore,  $\frac{1}{2}N(N + 1)$  distinct elements that need to be estimated from the training data. To avoid the matrix being singular at least  $N(N + 1)$  independent *samples* is needed. Fortunately, each  $N$  dimensional pixel vector in fact contains  $N$  samples (one in each waveband); thus the minimum number of independent training *pixels* required is  $(N + 1)$ . Because of the difficulty in assuring independence of the pixels, usually many more than this minimum number is selected. Swain and Davis (1978) recommend as a practical minimum that  $10N$  training pixels per spectral class be used, with as many as  $100N$  per class if possible. For data with low dimensionality (say up to 5 or 6 bands) those numbers can usually be achieved, but for hyperspectral data sets finding enough training pixels per class is extremely difficult. Section 13.5 considers this problem in some detail.

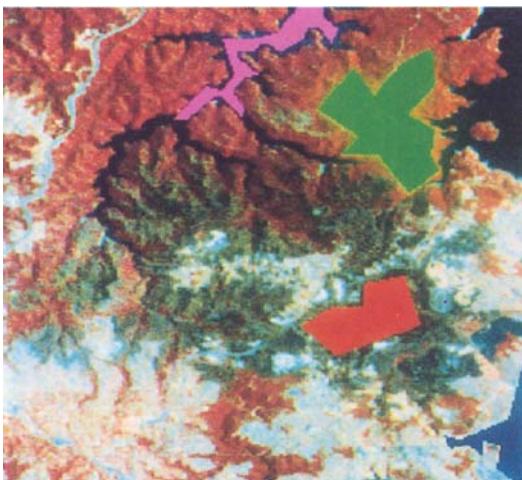
### 8.2.7

#### A Simple Illustration

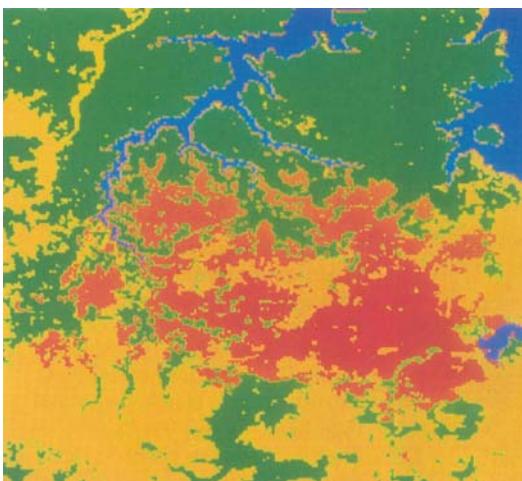
As an example of the use of maximum likelihood classification, the segment of Landsat multispectral scanner image shown in Fig. 8.3 is chosen. This is a  $256 \times 276$  pixel array of image data in which four broad ground cover types are evident. These are water, fire burn, vegetation and “developed” land (urban). Suppose we want to produce a thematic map of these four cover types in order to enable the area and extent of the fire burn to be evaluated.

The first step is to choose training data. For such a broad classification, suitable sets of training pixels for each of the four classes are easily identified visually in the image data. Figure 8.3 also shows the locations of four training fields used for this purpose. Sometimes, to obtain a good estimate of class statistics it may be necessary to choose several training fields for the one cover type, located in different regions of the image.

The four-band signatures for each of the four classes, as obtained from the training fields, are given in Table 8.1. The mean vectors can be seen to agree generally



**Fig. 8.3.** Image segment to be classified, consisting of a mixture of natural vegetation, waterways, urban development and vegetation damaged by fire. Four training regions are identified in solid colour. These are water (violet), vegetation (green), fire burn (red) and urban (dark blue in the bottom right hand corner). Pixels from these were used to generate the signatures in Table 8.1



**Fig. 8.4.** Thematic map produced by maximum likelihood classification. Blue represents water, red is fire damaged vegetation, green is natural vegetation and yellow is urban development

with known spectral reflectance characteristics of the cover types. Also the class variances (diagonal elements in the covariance matrices) are small for water as might be expected but on the large side for the developed/urban class, indicative of its heterogeneous nature.

Using these signatures in a maximum likelihood algorithm to classify the four bands of the image in Fig. 8.3, the thematic map shown in Fig. 8.4 is obtained. The four classes, by area, are given in Table 8.2. Note that there are no unclassified pixels, since a threshold was not used in the labelling process. The area estimates are obtained by multiplying the number of pixels per class by the effective area of a pixel. In the case of the Landsat 2 multispectral scanner the pixel size was 0.4424 hectares.

**Table 8.1.** Class signatures generated from the training areas in Fig. 8.3. Numbers are on a scale of 0 to 255 (8 bit)

Class	Mean vector	Covariance matrix			
Water	44.27	14.36	9.55	4.49	1.19
	28.82	9.55	10.51	3.71	1.11
	22.77	4.49	3.71	6.95	4.05
	13.89	1.19	1.11	4.05	7.65
Fire burn	42.85	9.38	10.51	12.30	11.00
	35.02	10.51	20.29	22.10	20.62
	35.96	12.30	22.10	32.68	27.78
	29.04	11.00	20.62	27.78	30.23
Vegetation	40.46	5.56	3.91	2.04	1.43
	30.92	3.91	7.46	1.96	0.56
	57.50	2.04	1.96	19.75	19.71
	57.68	1.43	0.56	19.71	29.27
Developed (urban)	63.14	43.58	46.42	7.99	-14.86
	60.44	46.42	60.57	17.38	-9.09
	81.84	7.99	17.38	67.41	67.57
	72.25	-14.86	-9.09	67.57	94.27

**Table 8.2.** Tabular summary of the thematic map of Fig. 8.4

Class	No. of pixels	Area (ha)
Water	4830	2137
Fireburn	14182	6274
Vegetation	28853	12765
Developed (urban)	22791	10083

## 8.3

### Minimum Distance Classification

#### 8.3.1

#### The Case of Limited Training Data

The effectiveness of maximum likelihood classification depends upon reasonably accurate estimation of the mean vector  $\mathbf{m}$  and the covariance matrix  $\Sigma$  for each spectral class. This in turn is dependent upon having a sufficient number of training pixels for each of those classes. In cases where this is not so, inaccurate estimates of the elements of  $\Sigma$  result, leading to poor classification. When the number of training samples per class is limited it can be more effective to resort to a classifier that does not make use of covariance information but instead depends only upon the mean positions of the spectral classes, noting that for a given number of samples these can be more accurately estimated than covariances. The so-called minimum distance classifier, or more precisely, minimum distance to class means classifier, is such an

approach. With this classifier, training data is used only to determine class means; classification is then performed by placing a pixel in the class of the nearest mean.

The minimum distance algorithm is also attractive since it is a faster technique than maximum likelihood classification, as will be seen in Sect. 8.5. However because it does not use covariance data it is not as flexible as the latter. In maximum likelihood classification each class is modelled by a multivariate normal class model that can account for spreads of data in particular spectral directions. Since covariance data is not used in the minimum distance technique class models are symmetric in the spectral domain. Elongated classes therefore will not be well modelled. Instead several spectral classes may need to be used with this algorithm where one might be suitable for maximum likelihood classification. This point is developed further in the case studies of Chap. 11.

### 8.3.2

#### The Discriminant Function

The discriminant function for the minimum distance classifier is developed as follows.

Suppose  $\mathbf{m}_i, i = 1, \dots, M$  are the means of the  $M$  classes determined from training data, and  $\mathbf{x}$  is the position of the pixel to be classified. Compute the set of squared Euclidean distances of the unknown pixel to each of the class means, defined in vector form as

$$\begin{aligned} d(\mathbf{x}, \mathbf{m}_i)^2 &= (\mathbf{x} - \mathbf{m}_i)^t (\mathbf{x} - \mathbf{m}_i) \\ &= (\mathbf{x} - \mathbf{m}_i) \cdot (\mathbf{x} - \mathbf{m}_i), i = 1, \dots, M \end{aligned}$$

Expanding the product gives

$$d(\mathbf{x}, \mathbf{m}_i)^2 = \mathbf{x} \cdot \mathbf{x} - 2\mathbf{m}_i \cdot \mathbf{x} + \mathbf{m}_i \cdot \mathbf{m}_i.$$

Classification is performed on the basis of

$$\mathbf{x} \in \omega_i \quad \text{if} \quad d(\mathbf{x}, \mathbf{m}_i)^2 < d(\mathbf{x}, \mathbf{m}_j)^2 \quad \text{for all } j \neq i$$

Note that  $\mathbf{x} \cdot \mathbf{x}$  is common to all  $d(\mathbf{x}, \mathbf{m}_j)^2$  and thus can be removed. Moreover, rather than classifying according to the smallest of the remaining expressions, the signs can be reversed and classification performed on the basis of

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all } j \neq i \tag{8.11a}$$

where

$$g_i(\mathbf{x}) = 2\mathbf{m}_i \cdot \mathbf{x} - \mathbf{m}_i \cdot \mathbf{m}_i, \quad \text{etc.} \tag{8.11b}$$

Equation (8.11b) defines the discriminant function for the minimum distance classifier. In contrast to the maximum likelihood approach the decision surfaces for this classifier, separating the distinct spectral class regions in multispectral space, are linear, as seen in Sect. 8.3.4 following. The higher order decision surface possible with maximum likelihood classification renders it more powerful for partitioning multispectral space than the linear surfaces for the minimum distance approach.

Nevertheless, as noted earlier, minimum distance classification is of value when the number of training samples is limited and, in such a case, can lead to better accuracies than the maximum likelihood procedure.

Minimum distance classification can be performed also using distance measures other than Euclidean (Wacker and Landgrebe, 1972); notwithstanding this, algorithms based upon Euclidean distance definitions are those generally implemented in software packages for remote sensing image analysis, such as Multispec (<http://dynamo.ecn.purdue.edu/~biehl/MultiSpec>), ENVI (<http://www.rsinc.com>) and ERMapper (<http://www.ermapper.com>).

### 8.3.3

#### **Degeneration of Maximum Likelihood to Minimum Distance Classification**

The major difference between the minimum distance and maximum likelihood classifiers lies in the use, by the latter, of the sample covariance information. Whereas the minimum distance classifier labels a pixel as belonging to a particular class on the basis only of its distance from the relevant mean, irrespective of its direction from that mean, the maximum likelihood classifier modulates its decision with direction, based upon the information in the covariance matrix. Furthermore the entry  $-\frac{1}{2} \ln |\Sigma_i|$  in its discriminant function shows explicitly that patterns have to be closer to some means than others to have equivalent likelihoods of class membership. As a result substantially superior performance is expected of the maximum likelihood classifier, in general. The following situation however warrants consideration since then there is no advantage in maximum likelihood procedures. It could occur in practice when class covariance is dominated by systematic noise rather than by natural spectral spreads of the individual spectral classes.

Consider the covariance matrices of all classes to be diagonal and equal, and the variances in each component to be identical, so that

$$\Sigma_i = \sigma^2 I \quad \text{for all } i.$$

Under these circumstances the discriminant function for the maximum likelihood classifier, from (8.7) becomes

$$g_i(\mathbf{x}) = \frac{1}{2} \ln \sigma^{2N} - \frac{1}{2\sigma^2} (\mathbf{x} - \mathbf{m}_i)^t (\mathbf{x} - \mathbf{m}_i) + \ln p(\omega_i)$$

The  $\ln \sigma^{2N}$  term is now common to all classes and can be ignored, as can the  $\mathbf{x} \cdot \mathbf{x}$  term that results from the scalar product, leaving

$$g_i(\mathbf{x}) = \frac{1}{2\sigma^2} \{2\mathbf{m}_i \cdot \mathbf{x} - \mathbf{m}_i \cdot \mathbf{m}_i\} + \ln p(\omega_i)$$

If the  $\ln p(\omega_i)$  are ignored, on the basis of equal prior probabilities, then the  $1/2\sigma^2$  factor can be removed giving

$$g_i(\mathbf{x}) = 2\mathbf{m}_i \cdot \mathbf{x} - \mathbf{m}_i \cdot \mathbf{m}_i$$

which is the discriminant function for the minimum distance classifier. Thus minimum distance and maximum likelihood classification are equivalent for identical and symmetric spectral class distributions.

### 8.3.4 Decision Surfaces

The implicit surfaces in multispectral space separating adjacent classes are defined by the respective discriminant functions being equal. Thus the surface between the  $i$ th and  $j$ th spectral classes is given by

$$g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0$$

Substituting from (8.11b) gives

$$2(\mathbf{m}_i - \mathbf{m}_j) \cdot \mathbf{x} - (\mathbf{m}_i \cdot \mathbf{m}_i - \mathbf{m}_j \cdot \mathbf{m}_j) = 0$$

This defines a linear surface – often called a hyperplane in more than three dimensions. In contrast therefore to maximum likelihood classification in which the decision surfaces are quadratic and therefore more flexible, the decision surfaces for minimum distance classification are linear and more restricted.

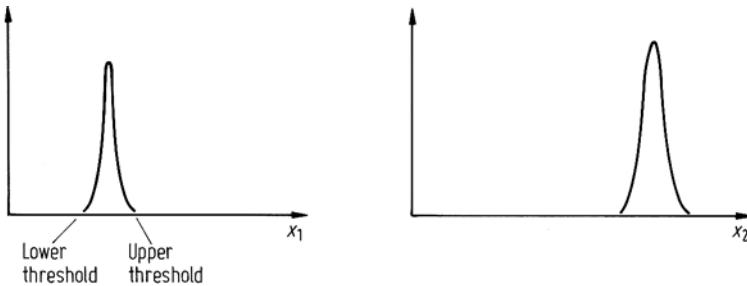
### 8.3.5 Thresholds

Thresholds can be applied to minimum distance classification by ensuring that not only is a pixel closest to a candidate class but also that it is within a prescribed distance of that class. Such a technique is used regularly. Often the distance threshold is specified according to a number of standard deviations from a class mean.

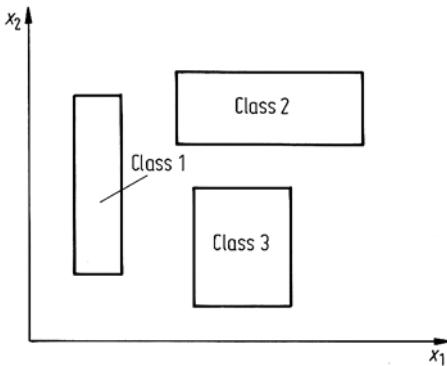
## 8.4 Parallelepiped Classification

The parallelepiped classifier is a very simple supervised classifier that is, in principle, trained by inspecting histograms of the individual spectral components of the available training data. Suppose, for example, that the histograms of one particular spectral class for two dimensional data are as shown in Fig. 8.5. Then the upper and lower significant bounds on the histograms are identified and used to describe the brightness value range for each band for that class. Together, the range in all bands describes a multidimensional box or parallelepiped. If, on classification, pixels are found to lie in such a parallelepiped they are labelled as belonging to that class. A two-dimensional pattern space might therefore be segmented as shown in Fig. 8.6.

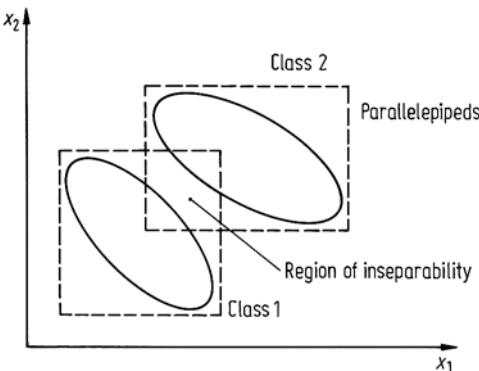
While the parallelepiped method is, in principle, a particularly simple classifier to train and use, it has several drawbacks. One is that there can be considerable gaps between the parallelepipeds; pixels in those regions will not be classified. By



**Fig. 8.5.** Histograms for the components of a two-dimensional set of training data corresponding to a single spectral class. The upper and lower bounds are identified as the edges of a two-dimensional parallelepiped



**Fig. 8.6.** An example of a set of two-dimensional parallelepipeds



**Fig. 8.7.** Parallelepiped classification of correlated data showing regions of inseparability

comparison the minimum distance and maximum likelihood classifiers will label all pixels in an image, unless thresholding methods are used. Another limitation is that prior probabilities of class membership are not taken account of; nor are they for minimum distance classification. Finally, for correlated data there can be overlap of the parallelepipeds since their sides are parallel to the spectral axes. Consequently there is some data that cannot be separated, as illustrated in Fig. 8.7.

## 8.5

### Classification Time Comparison of the Classifiers

Of the three classifiers commonly used with remote sensing image data the parallelepiped procedure is the fastest in classification since only comparisons of the spectral components of a pixel with the spectral dimensions of the parallelepipeds are required.

For the minimum distance classifier the discriminant function in (8.11b) requires evaluation for each pixel. In practice  $2\mathbf{m}_i$  and  $\mathbf{m}_i \cdot \mathbf{m}_i$  would be calculated beforehand, leaving  $N$  multiplications and  $N$  additions to check the potential membership of a pixel to one class, where  $N$  is the number of components in  $\mathbf{x}$ .

By comparison, evaluation of the discriminant function for maximum likelihood classification in (8.7) requires  $N^2 + N$  multiplications and  $N^2 + 2N + 1$  additions, to check one pixel against one class, given that

$$-\frac{1}{2} \ln |\Sigma_i| + \ln p(\omega_i)$$

would have been calculated beforehand. Ignoring additions by comparison to multiplications, the maximum likelihood classifier takes  $N + 1$  times as long as the minimum distance classifier to perform a classification. It is also significant to note that classification time, and thus cost, increases quadratically with number of spectral components for the maximum likelihood classifier but only linearly for minimum distance and parallelepiped classification. This has particular relevance to feature reduction (Chap. 10).

## 8.6

### Other Supervised Approaches

#### 8.6.1

##### The Mahalanobis Classifier

Consider the discriminant function for the maximum likelihood classifier, for the special case of equal prior probabilities, as defined in (8.8). If the sign of this function is reversed it can be considered as a distance squared measure since the quadratic entry has those dimensions and the other term is a constant. Thus we can define

$$d(\mathbf{x}, \mathbf{m}_i)^2 = \ln |\Sigma_i| + (\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1} (\mathbf{x} - \mathbf{m}_i) \quad (8.12)$$

and classify on the basis of the smallest  $d(\mathbf{x}, \mathbf{m}_i)$  as for the Euclidean minimum distance classifier. Thus the maximum likelihood classifier can be regarded as a minimum distance-like classifier but with a distance measure that is direction sensitive and modified according to class.

Consider the case now where all class covariances are equal – i.e.  $\Sigma_i = \Sigma$  for all  $i$ . Clearly the  $\ln |\Sigma_i|$  term is now not discriminating and can be ignored. The

distance measure then reduces to

$$d(\mathbf{x}, \mathbf{m}_i)^2 = (\mathbf{x} - \mathbf{m}_i)^t \Sigma^{-1} (\mathbf{x} - \mathbf{m}_i) \quad (8.13)$$

Such a classifier is referred to as a *Mahalanobis distance* classifier, although sometimes the term is applied to the more general measure of (8.12). Mahalanobis distance is understood as the square root of (8.13). Under the additional constraint that  $\Sigma = \sigma^2 I$  the Mahalanobis classifier reduces, as before, to the minimum Euclidean distance classifier.

The advantage of the Mahalanobis classifier over the maximum likelihood procedure is that it is faster and yet retains a degree of direction sensitivity via the covariance matrix  $\Sigma$ , which could be a class average or a pooled variance.

### 8.6.2

#### Table Look Up Classification

Since the set of discrete brightness values that can be taken by a pixel in each spectral band is limited, there is a finite, although large, number of pixel vectors in any particular image. For a given class in that image the number of distinct pixel vectors may not be very extensive. Consequently a viable classification scheme is to note the set of pixel vectors corresponding to a given class, using representative training data, and then use those to classify the image by comparing unknown image pixels with each pixel in the training data until a match is found. No arithmetic operations are required and, notwithstanding the number of comparisons that might be necessary to determine a match, it is a fast classifier. It is referred to as a look up table approach since the pixel brightnesses are stored in tables that point to the corresponding classes.

An obvious drawback with this approach is that the chosen training data must contain one of every possible pixel vector for each class. Should some be missed then the corresponding pixels in the image will be left unclassified. This is in contrast to the procedures treated above.

### 8.6.3

#### The $kNN$ (Nearest Neighbour) Classifier

A classifier that is particularly simple in concept, but can be time consuming to apply, is the  $k$ -Nearest Neighbour classifier. It assumes that pixels close to each other in feature space are likely to belong to the same class. In its simplest form an unknown pixel is labelled by examining the available training pixels in multispectral space and choosing the class most represented among a pre-specified number of nearest neighbours. The comparison essentially requires the distances from the unknown pixel to all training pixels to be computed.

Suppose there are  $k_i$  neighbours labelled as class  $\omega_i$  out of  $k$  nearest neighbours for a pixel vector  $\mathbf{x}$ , noting that  $\sum_{i=1}^M k_i = k$  where  $M$  is the total number of classes.

In the basic  $kNN$  rule we define the discriminant function for the  $i$ th class as

$$g_i(\mathbf{x}) = k_i$$

and the decision rule is:

$$\mathbf{x} \in \omega_i, \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i$$

The basic rule does not take the distance of each neighbour to the current pixel vector into account and may lead to tied results. An improvement is to distance-weight the discriminant function:

$$g_i(\mathbf{x}) = \frac{\sum_{j=1}^{k_i} 1/d(\mathbf{x}, \mathbf{x}_i^j)}{\sum_{i=1}^M \sum_{j=1}^{k_i} 1/d(\mathbf{x}, \mathbf{x}_i^j)}$$

where  $d(\mathbf{x}, \mathbf{x}_i^j)$  is the spectral distance (commonly Euclidean) between the unknown pixel vector  $\mathbf{x}$  and its neighbour  $\mathbf{x}_i^j$ , the  $j$ th of the  $k_i$  pixels in class  $\omega_i$ .

If the training data for each class is not in proportion to its respective population,  $p(\omega_i)$ , in the image, a Bayesian Nearest-Neighbour rule can be used:

$$g_i(\mathbf{x}) = \frac{p(\mathbf{x} | \omega_i) p(\omega_i)}{\sum_{j=1}^M p(\mathbf{x} | \omega_j) p(\omega_j)} = \frac{k_i p(\omega_i)}{\sum_{j=1}^M k_j p(\omega_j)}$$

In the  $kNN$  algorithm as many spectral distances as there are training pixels must be evaluated for each training pixel to be labelled; that requires an impractically high computational load, especially when the number of spectral bands and/or the number of training samples is large. The method is not well-suited therefore to hyperspectral data sets, although it is possible to improve the efficiency of the distance search process (Dasarathy, 1991).

## 8.7 Gaussian Mixture Models

In Sect. 3.5 the prospect of information classes being composed of sets of spectral classes was raised. The material in Sect. 8.2, however, has been derived on the basis that an information class is composed of a single spectral class that can be represented by a multidimensional normal distribution. In practice that is rarely the case: in order to represent the data effectively more than one normally distributed spectral class is required to model properly the distribution of pixel vectors in a given information class.

One of the challenges to successful image classification is to find an acceptable set of spectral classes for each information class. In Sect. 9.1 it is suggested that clustering algorithms can be used for that purpose, and indeed they can be applied

very successfully to that end. Another, more theoretically appealing approach is to try to learn the mixture of spectral classes for each information class from the available training data, as in the following.

If we assume that a given information class is composed of a number of unimodal normally distributed spectral classes, then it is natural to attempt to devise a (information) class model of the form

$$f(\mathbf{x}) = \sum_{c=1}^C \alpha_c p(\mathbf{x} | \mathbf{m}_c, \Sigma_c)$$

where  $\mathbf{m}_c$  and  $\Sigma_c$  are the mean vector and covariance matrix of the  $c$ th spectral class conditional normal distribution; the  $\alpha_c$  are weighting parameters (which sum to unity) such that the mixture model expressed by  $f(\mathbf{x})$  fits the available training data. The total number of spectral class components is  $C$ .

We have to estimate the set of parameters  $\{C, \alpha_c, m_c, \Sigma_c\}$  from the training data, and that is a considerable challenge in practice. Kuo and Landgrebe (2002) show how this can be achieved.

## 8.8 Context Classification

### 8.8.1

#### The Concept of Spatial Context

The classifiers treated so far are often referred to as point or pixel-specific classifiers in that they label a pixel on the basis of its spectral properties alone, with no account taken of how any neighbouring pixels are labelled. Yet, in any real image, adjacent pixels are related or correlated, both because imaging sensors acquire significant portions of energy from adjacent pixels<sup>1</sup> and because ground cover types generally occur over a region that is large compared with the size of a pixel. In an agricultural area, for example, if a particular image pixel represents wheat it is highly likely that its neighbouring pixels will also be wheat. This knowledge of neighbourhood relationships is a rich source of information that is not exploited in simple, traditional classifiers. In this section we consider the importance of spatial context and see the benefit of taking it into account when making classification decisions. Not only is the inclusion of context important because it exploits spatial information, as such, but, in addition, sensitivity to the correct context for a pixel can improve a thematic map by helping to remove individual pixel labelling errors that might result from noisy data, or unusual classifier performance (see Problem 8.6).

Classification methods that take into account the labelling of neighbours when seeking to determine the most appropriate class for a pixel are said to be context sensitive, or simply context classifiers. They attempt to develop a thematic map that is consistent both spectrally and spatially.

---

<sup>1</sup> This is referred to as the point spread function effect, which is discussed in Forster (1982).

The degree to which adjacent pixels are strongly correlated will depend on the spatial resolution of the sensor and the scale of natural and cultural regions on the earth's surface. Adjacent pixels over an agricultural region will be strongly correlated, whereas for the same sensor, adjacent pixels over a busier, urban region would not show strong correlation. Likewise, for a given area, neighbouring Landsat MSS pixels, being larger, may not demonstrate as much correlation as adjacent SPOT HRV pixels. In general terms, context classification techniques usually warrant consideration when processing higher resolution imagery.

### 8.8.2

#### Context Classification by Image Pre-processing

Perhaps the simplest method for exploiting spatial context is to process the image data before classification in order to modify or enhance its spatial properties. A median filter (Sect. 5.5.2), for example, will help in reducing salt and pepper noise that would lead to inconsistent class labels. Moreover, the application of simple averaging filters (possibly with edge preserving thresholds) can be used to impose a degree of homogeneity among the brightness values of adjacent pixels thereby increasing the chance that neighbouring pixels may be given the same label.

An alternative is to generate a separate channel of data that associates spatial properties with pixels. For example, a texture channel could be added and classification carried out (using a suitable algorithm such as the minimum distance rule) on the combined multispectral and texture channels. Along this line, Gong and Howarth (1990) have set up a "structural information" channel to bias a classification according to the density of high spatial frequency data in order to improve the classification of image data containing urban segments. The reasoning behind the approach is that urban regions are characterised by high spatial frequency detail whereas, conversely, the high frequency detail present in non-urban regions is low. The additional channel reflects this understanding and accordingly influences the classification which would otherwise be carried out on the basis of spectral data alone.

One of the more useful spatial pre-processing techniques is that used in the ECHO classification methodology. In ECHO (Extraction and Classification of Homogeneous Objects) regions of similar spectral properties are "grown" before classification is performed. Several region growing techniques are available, possibility the simplest of which is to aggregate pixels into small regions by comparing their brightnesses in each channel and then aggregate the small regions into bigger regions in a similar manner. When this is done, ECHO classifies the regions as single objects and only resorts to standard maximum likelihood classification when it has to treat individual pixels that could not be put into regions. Details of ECHO will be found in Kettig and Landgrebe (1976); it is also available in the Multispec image analysis software (<http://dynamo.ecn.purdue/~biehl/Multispec/>).

### 8.8.3

#### Post Classification Filtering

Once a thematic map has been generated using a simple point classifier some degree of spatial context can be developed by logically filtering the map. For example, if the map is examined in  $3 \times 3$  windows, a label at the centre of the window might be changed to the label most represented in the window. Clearly this must be done carefully, with the user having some control over the minimum size region of a given cover type that is acceptable in the filtered image product (Harris, 1985). Post classification filtering by this approach has been treated by Townsend (1986).

### 8.8.4

#### Probabilistic Label Relaxation

Spatial consistency in a classified image product can also be developed using the process of label relaxation. While it has little theoretical foundation, and is more complex than the methods outlined in the previous sections, it does allow the spatial properties of a region to be carried into the classification process in a logically consistent way.

##### 8.8.4.1

###### The Basic Algorithm

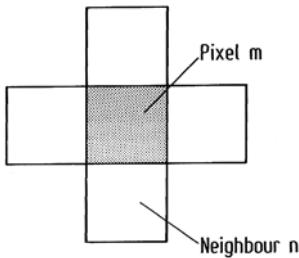
The process commences by assuming that a classification, based on spectral data alone, has already been carried out. There is available therefore, for each pixel, a set of probabilities that describe the chance that the pixel belongs to each of the possible ground cover classes under consideration. This set of probabilities could be computed from (8.6) and (8.7) if maximum likelihood classification had been used first. If another classification method had been employed, then some other assignment process will be required. It could even be as simple as allocating a high probability to the most favoured class label and lower probabilities to the rest. Let the set of probabilities for a pixel ( $m$ ) currently of interest be represented by

$$p_m(\omega_i) \quad i = 1, \dots, K \quad (8.14)$$

where  $K$  is the total number of classes;  $p_m(\omega_i)$  should be read as “the probability that  $\omega_i$  is the correct class for pixel  $m$ .” Note that the full set of  $p_m(\omega_i)$  must sum to unity for a given pixel – viz.

$$\sum_i p_m(\omega_i) = 1.$$

Suppose now that a neighbourhood is defined surrounding pixel  $m$ . This can be of any size and, in principle, should be large enough to ensure that all the pixels considered to have any spatial correlation with  $m$  are included. For high resolution imagery this is not practical and simple neighbourhoods such as that shown in Fig. 8.8 are often adopted.



**Fig. 8.8.** Definition of a simple neighbourhood about pixel  $m$

Now assume that a *neighbourhood function*  $Q_m(\omega_i)$  can be found (by means to be described below) which allows the pixels in the prescribed neighbourhood to influence the possible classification of pixel  $m$ . This influence is exerted by multiplying the label probabilities in (8.14) by the  $Q_m(\omega_i)$ . However, so that the new set of label probabilities sum to one, these new values are divided by their sum:

$$p'_m(\omega_i) = \frac{p_m(\omega_i)Q_m(\omega_i)}{\sum_i p_m(\omega_i)Q_m(\omega_i)} \quad (8.15)$$

Such a modification is made to the set of label probabilities for all pixels by moving over the image from its top left hand to bottom right hand corners. In the following it will be seen that the neighbourhood function  $Q_m(\omega_i)$  depends on the label probabilities of the neighbouring pixels, so that if all the pixel probabilities are modified in the manner just described then the neighbours for any given pixel have also been altered. Consequently, (8.15) should be applied again to give newer estimates still of the label probabilities. Indeed, (8.15) is applied as many times as necessary to ensure that the  $p'_m(\omega_i)$  have stabilised – i.e. that they do not change with further iteration. It is assumed that the  $p'_m(\omega_i)$  then represent the correct set of label probabilities for the pixel, having taken account both of spectral data (in the initial determination of label probabilities) and spatial context (via the neighbourhood functions). Since the process is iterative, (8.15) is usually written as an explicit iteration formula:

$$p_m^{k+1}(\omega_i) = \frac{p_m^k(\omega_i)Q_m^k(\omega_i)}{\sum_i p_m^k(\omega_i)Q_m^k(\omega_i)} \quad (8.16)$$

where  $k$  is the iteration counter. Depending on the size of the image and its spatial complexity, the number of iterations required to stabilise the label probabilities may be quite large. However, most change in the label probabilities occurs in the first few iterations and there is good reason to believe that proceeding beyond say 5 to 10 iterations may not be necessary in most cases (see Sect. 8.8.4.4).

### 8.8.4.2 The Neighbourhood Function

Consider just one of the neighbours of pixel  $m$  in Fig. 8.8 – call it pixel  $n$ . Suppose there is available a measure of compatibility of the current labelling of pixel  $m$  and

its neighbouring pixel  $n$ . For example let  $r_{mn}(\omega_i, \omega_j)$  describe numerically how compatible it is to have pixel  $m$  classified as  $\omega_i$  and neighbouring pixel  $n$  classified as  $\omega_j$ . It would be expected, for example, that this measure will be high if the adjoining pixels are both labelled wheat in an agricultural region, but low if one of the neighbours was classified as snow. There are several ways these *compatibility coefficients*, as they are called, can be defined. An intuitively appealing definition is based on conditional probabilities. Thus, the compatibility measure  $p_{mn}(\omega_i | \omega_j)$  is the probability that  $\omega_i$  is the correct label for pixel  $m$  if  $\omega_j$  is the correct label on pixel  $n$ . A small piece of evidence in favour of  $\omega_i$  being correct for pixel  $m$  is  $p_{mn}(\omega_i | \omega_j) p_n(\omega_j)$  – i.e. the probability that  $\omega_i$  is correct for pixel  $m$  if  $\omega_j$  is correct for pixel  $n$  multiplied by the probability that  $\omega_j$  is correct for pixel  $n$ <sup>2</sup>. Since probabilities for all possible labels on pixel  $n$  are available (even though some might be very small) the total evidence from pixel  $n$  in favour of  $\omega_i$  being the correct class for pixel  $m$  will be the sum of the contributions from all pixel  $n$ 's labelling possibilities, viz.

$$\sum_j p_{mn}(\omega_i | \omega_j) p_n(\omega_j).$$

Consider now the full neighbourhood of the pixel  $m$ . In a like manner all the neighbours contribute evidence in favour of labelling pixel  $m$  as coming from class  $\omega_i$ . All these contributions are simply added<sup>3</sup>, via the use of *neighbour weights*  $d_n$  that recognise that some neighbours may be more influential than others (as for example, pixels along a scan line in MSS data compared with those running down an image, owing to the oversampling that occurs along rows – see Fig. A.2). Thus, at the  $k$ th iteration, the total neighbourhood support for pixel  $m$  being classified as  $\omega_i$  is:

$$Q_m^k(\omega_i) = \sum_n d_n \sum_j p_{mn}(\omega_i | \omega_j) p_n^k(\omega_j) \quad (8.17)$$

This is the definition of the neighbourhood function. In (8.16) and (8.17) it is common to include pixel  $m$  in its own neighbourhood so that the modification process is not entirely dominated by the neighbours, particularly if the number of iterations is so large as to take the process quite a long way from its starting point.

Unless there is good reason to do otherwise the neighbour weights are generally chosen all to be the same.

#### 8.8.4.3

#### Determining the Compatibility Coefficients

Several methods are possible for determining values for the compatibility coefficients  $p_{mn}(\omega_i | \omega_j)$ . One is to have available a spatial model for the region under consideration, derived from some other data source. In an agricultural region, for example,

<sup>2</sup> This is the probability of the joint event that pixel  $m$  is labelled  $\omega_i$  and pixel  $n$  is labelled  $\omega_j$ .

<sup>3</sup> An alternative way of handling the full neighbourhood is to take the geometric mean of the neighbourhood contributions.

some general idea of field sizes along with a knowledge of the pixel size of the sensor being used should make it possible to estimate how often one particular class occurs following a given class on an adjacent pixel. Another approach is to compute values for the compatibility coefficients from ground truth pixels, although the ground truth needs to be in the form of training regions that contain heterogeneous and spatially representative cover types.

#### **8.8.4.4**

#### **The Final Step – Stopping the Process**

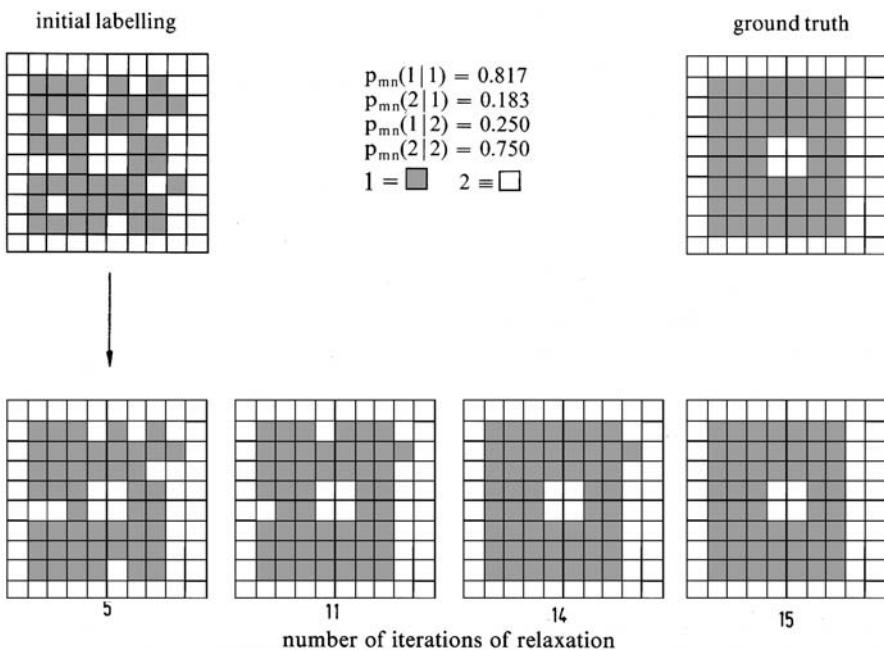
While the relaxation process operates on label probabilities, the user is interested in the actual labels themselves. At the completion of relaxation, or at any intervening stage, each of the pixels can be classified according to the highest label probability. Thought has to be given as to how and when the iterations should be terminated. As suggested earlier, the process can be allowed to go to a natural completion at which further iteration leads to no changes in the label probabilities for all pixels. This however presents two difficulties. First, up to several hundred iterations may be involved leading to a costly post classification step. Secondly, it is observed in practice that the relaxation process improves the classification results in the first few iterations, by the embedding of spatial information, often to deteriorate later in the process (Richards, Landgrebe and Swain, 1981). Indeed, if the process is not terminated, the thematic map, after a large number of iterations of relaxation, can be worse than before the technique was applied.

To avoid these difficulties, a stopping rule or other controlling mechanism is needed. As seen in the example of the following section, stopping after just a few iterations may allow most of the benefit to be drawn from the process. Alternatively, the labelling errors remaining at each iteration can be checked against ground truth, if available, and the iterations terminated when the labelling error is seen to be minimised (Gong and Howarth, 1989).

Another approach is to control the propagation of contextual information as iteration proceeds (Lee, 1984). A little thought will reveal that, in the first iteration, only the immediate neighbours of a pixel have an influence on its labelling. In the second iteration the neighbours two away will now have an influence via the intermediary of the intervening pixels. Similarly, as iterations proceed, information from neighbours further away is propagated into the pixel of interest to modify its label probabilities. If the user has a view of the separation between neighbours at which the spatial correlation has dropped to negligible levels, then the appropriate number of iterations should be able to be identified at which to terminate the process without unduly sacrificing any further improvement in labelling accuracy. Noting also that the nearest neighbours should be most influential, with those further out being less important, a useful variation is to reduce the values of the neighbour weights  $d_n$  as iteration proceeds so that after say 5 to 10 iterations they have been brought to zero. Further iterations will then have no effect, and degradation in labelling accuracy cannot occur (Lee and Richards, 1989).

### 8.8.4.5 Examples

Figure 8.9 illustrates a simple application of relaxation labelling, in which a hypothetical image of 100 pixels has been classified into just two classes – grey and white. The ground truth for the region is shown, along with the thematic map (initial labelling) assumed to have been generated from a point classifier such as the maximum likelihood rule. Also shown are the compatibility coefficients, expressed as conditional probabilities, computed from the ground truth map. Label probabilities were assumed to be 0.9 for the favoured label in the initial labelling and 0.1 for the less likely label. The initial labelling, by comparison with the ground truth, can be seen to have an accuracy of 82% (there are 12 pixels in error). The labelling (selected on the basis of the largest current label probability) at significant stages during iteration is shown, illustrating the reduction in classification error resulting from the incorporation of spatial information into the process. After 15 iterations all initial labelling errors have been removed, leading to a thematic map 100% in agreement with the ground truth. In this case the relaxation process was allowed to proceed to completion and there have been no ill effects. As pointed out in the previous section, however, this is the exception and stopping rules may have to be applied in most cases. Other simple examples where this is the case will be found in Richards et al., (1981).



**Fig. 8.9.** Simple demonstration of pixel relaxation labelling

As a second example, the leftmost  $82 \times 100$  pixels of the agricultural image shown in Fig. 3.1 have been chosen. Figure 8.10a shows the ground truth for the image segment and Fig. 8.10b shows the result of a maximum likelihood classification. The initial classification accuracy is 65.6%. The relaxation process was initialised using actual probability estimates from the maximum likelihood rule. Conditional probabilities as such were not used as compatibility coefficients. Instead, a slightly different set of compatibilities as proposed by Peleg and Rosenfeld (1980) was adopted. Also, to control the propagation of context information and thereby obviate any deleterious effect of allowing the relaxation process to proceed unconstrained, the neighbourhood weights were diminished with iteration count as described in previous section. Figure 8.10c shows the final labelling, which has an accuracy of 72.4%. Full details of this example are available in Lee and Richards (1989).

### 8.8.5

#### Handling Spatial Context by Markov Random Fields

The effect of spatial context can also be incorporated into a classification using the concept of the Markov Random Field (MRF). It is useful in developing the Markov Random Field approach to commence by considering the whole image, rather than just a local neighbourhood. We will restrict our attention to a neighbourhood once we have established some fundamental concepts.

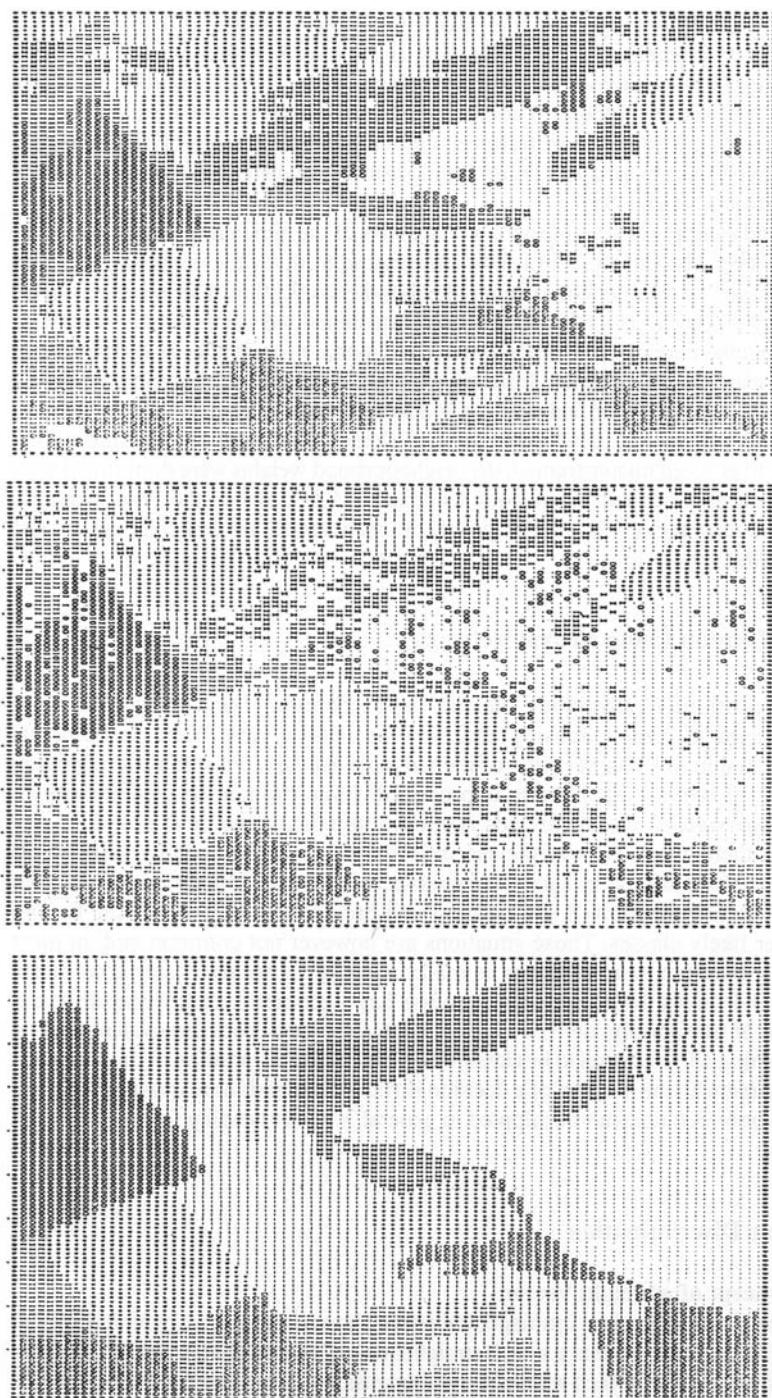
Suppose there is a total of  $M$  pixels in the image to be classified, with measurement vectors  $\mathbf{x}_1, \dots, \mathbf{x}_M$ . Alternatively, the measurement vectors can be expressed  $\{\mathbf{x}_m : m = 1, \dots, M\}$ , in which  $m \equiv (i, j)$  in our usual way of indexing the pixels in an image. We can describe the full set of measurement vectors by  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ . Further, suppose the class labels on each of the  $M$  pixels can be represented by the set  $\Omega = \{\omega_{c1}, \dots, \omega_{cM}\}$ ; we could refer to that as the *scene labelling*, because it looks at the classification of every pixel in the scene. Each  $\omega_{cm}$  can be one of  $c = 1, \dots, C$  available classes. By classification what we want to find, of course, is the scene labelling (or our best estimate) that matches the ground truth – i.e. the actual classes of the pixels on the earth’s surface. Let the actual labels on the ground be represented by  $\Omega^*$ .

There will be a probability distribution  $p(\Omega)$  associated with the labelling  $\Omega$  of the whole scene which describes the likelihood of finding that distribution of labels over the image.  $\Omega$  is sometimes referred to as a *random field*.

In principle, what we would like to do is find the scene labelling  $\hat{\Omega}$  – that is the classification of all pixels – that maximises the global posterior probability  $p(\Omega|X)$ , the probability that  $\Omega$  is the correct overall scene labelling given that the full set of measurement vectors for the scene is  $X$ . By using Bayes’ theorem we can express this as

$$\hat{\Omega} = \arg \max_{\Omega} \{p(X|\Omega)p(\Omega)\} \quad (8.18)$$

in which the argmax function says that we choose the value of  $\Omega$  that maximises its argument. The distribution  $p(\Omega)$  is the prior probability of the scene labelling.



**Fig. 8.10.** **a** Ground truth for the left-hand side of the image in Fig. 3.1. The symbols are: · = red soil, \* = bare soil (low moisture), I = dry bare soil, + = early vegetation growth, X = mixed bare soil, - = bare soil (moist or ploughed). **b** Result of a maximum likelihood classification of Landsat MSS data. **c** Result of applying relaxation labelling to the result in **b**, incorporating a reduction in the neighbour weights with iteration

What we need to do now essentially is to perform the maximisation in (8.18), recognising however that the pixels are contextually dependent ie. there is some spatial correlation among them because adjacent pixels are likely to come from the same class. To render the problem tractable we consider the posterior probability just at the individual pixel level, so that our objective, for pixel  $m$ , is to find the class  $c$  that maximises  $p(\omega_{cm}|\mathbf{x}_m, \omega_{\partial m})$  where  $\omega_{\partial m}$  is the labelling on the pixels in a neighbourhood about pixel  $m$ . A possible neighbourhood is that shown in Fig.8.8, although often the immediately diagonal neighbours about  $m$  can also be included. Now we note

$$\begin{aligned} p(\omega_{cm}|\mathbf{x}_m, \omega_{\partial m}) &= p(\mathbf{x}_m, \omega_{\partial m}, \omega_{cm})/p(\mathbf{x}_m, \omega_{\partial m}) \\ &= p(\mathbf{x}_m|\omega_{\partial m}, \omega_{cm})p(\omega_{\partial m}, \omega_{cm})/p(\mathbf{x}_m, \omega_{\partial m}) \\ &= p(\mathbf{x}_m|\omega_{\partial m}, \omega_{cm})p(\omega_{cm}|\omega_{\partial m})p(\omega_{\partial m})/p(\mathbf{x}_m, \omega_{\partial m}) \end{aligned}$$

The first term on the right hand side is similar to the class conditional distribution function, but conditional also on the neighbourhood labelling. It is reasonable to assume that the class conditional density is independent of the neighbourhood labelling so that  $p(\mathbf{x}_m|\omega_{\partial m}, \omega_{cm}) = p(\mathbf{x}_m|\omega_{cm})$ . Note also that the measurement vector  $\mathbf{x}_m$  and the neighbourhood labelling are independent of each other so that  $p(\mathbf{x}_m, \omega_{\partial m}) = p(\mathbf{x}_m)p(\omega_{\partial m})$ , so that the last expression becomes

$$\begin{aligned} p(\omega_{cm}|\mathbf{x}_m, \omega_{\partial m}) &= p(\mathbf{x}_m|\omega_{cm})p(\omega_{cm}|\omega_{\partial m})p(\omega_{\partial m})/p(\mathbf{x}_m)p(\omega_{\partial m}) \\ &= p(\mathbf{x}_m|\omega_{cm})p(\omega_{cm}|\omega_{\partial m})/p(\mathbf{x}_m) \end{aligned}$$

Since  $1/p(\mathbf{x}_m)$  does not contribute to the decision concerning the correct label for pixel  $m$  it can be removed from the last expression, leaving

$$p(\omega_{cm}|\mathbf{x}_m, \omega_{\partial m}) \propto p(\mathbf{x}_m|\omega_{cm})p(\omega_{cm}|\omega_{\partial m}) \quad (8.19)$$

Now consider the probability  $p(\omega_{cm}|\omega_{\partial m})$ . Essentially it is the probability that the correct class for pixel  $m$  is  $c$  given the classes currently on the neighbours of pixel  $m$ . In many ways it is analogous to the neighbourhood function for probabilistic relaxation in (8.17). It is also a conditional prior probability – i.e. a prior probability for the class on pixel  $m$  conditional on its neighbourhood. Because of this conditionality, the random fields of labels we are considering are now referred to as *Markov Random Fields (MRF)*.

The question is how do we now find a value for  $p(\omega_{cm}|\omega_{\partial m})$ ? It is a property of MRFs that we can express the conditional prior distribution in the form of a Gibbs distribution

$$p(\omega_{cm}|\omega_{\partial m}) = \frac{1}{Z} \exp\{-U(\omega_{cm})\} \quad (8.20a)$$

in which (based on the so-called Ising model)

$$U(\omega_{cm}) = \sum_{\partial m} \beta[1 - \delta(\omega_{cm}, \omega_{\partial m})] \quad (8.20b)$$

where  $\delta(\omega_{cm}, \omega_{\partial m})$  is the Kronecker delta, which is unity if the arguments are equal and zero otherwise;  $\beta > 0$  is a parameter with value fixed by the user when applying the MRF technique to control the influence of the neighbours.

Equation (8.20) is now substituted into (8.19) to generate a posterior probability that depends on the class conditional probability found from the available spectral measurements (the first term on the right hand side) and the effect of the spatial neighbourhood. However, as with (8.4), it is convenient to take the logarithm of (8.19) to yield (with the choice of  $Z = 1$ ), an MRF-based discriminant function for the class on pixel  $m$  assuming a multivariate normal class conditional density function:

$$\begin{aligned} g_{cm}(\mathbf{x}_m) = & -\frac{1}{2} \ln |\Sigma_c| - \frac{1}{2} (\mathbf{x}_m - \mathbf{m}_c) \Sigma_c^{-1} (\mathbf{x}_m - \mathbf{m}_c)^t \\ & - \sum_{\partial m} \beta [1 - \delta(\omega_{cm}, \omega_{\partial m})] . \end{aligned}$$

Recall that classification is carried out on the basis of finding the class for the pixel that maximises the discriminant function. Noting the negative signs above, the most appropriate class for pixel  $m$  can be found by minimising the expression

$$\begin{aligned} d_{cm}(\mathbf{x}_m) = & \frac{1}{2} \ln |\Sigma_c| + \frac{1}{2} (\mathbf{x}_m - \mathbf{m}_c) \Sigma_c^{-1} (\mathbf{x}_m - \mathbf{m}_c)^t \\ & + \sum_{\partial m} \beta [1 - \delta(\omega_{cm}, \omega_{\partial m})] \end{aligned} \quad (8.21)$$

To use (8.21) there needs to be an allocation of classes over the scene before the last term can be computed. Accordingly, an initial classification would be performed, say with the maximum likelihood classifier of Sect. 8.2.3. Equation (8.21) would then be used to modify the labels attached to the individual pixels to incorporate the effect of context. However, in so doing some (or initially many) of the labels on the pixels will be modified. The process should then be run again, and indeed as many times presumably until there are no further changes.

## 8.9

### Non-parametric Classification: Geometric Approaches

Statistical classification algorithms are the most commonly encountered labelling techniques used in remote sensing and, for this reason, have been the principal methods treated in this chapter. One of the valuable aspects of a statistical approach is that a *set* of relative likelihoods is produced. Even though, in the majority of cases, the *maximum* of the likelihoods is chosen to indicate the most probable label for a pixel, there remains nevertheless information in the remaining likelihoods that could be made use of in some circumstances, either to initiate a process such as relaxation labelling (Sect. 8.8.4) or simply to provide the user with some feeling for the other likely classes. Those situations are however not common and, in most applications, the maximum selection is made. That being so, the material in Sects. 8.2.4 and 8.3.4

shows that the decision process has a geometric counterpart in that a comparison of statistically derived discriminant functions leads equivalently to a decision rule that allows a pixel to be classified on the basis of its position in multispectral space compared with the location of a decision surface. This leads us to question whether a geometric interpretation can be adopted in general, without needing first to use statistical models.

### 8.9.1

#### Linear Discrimination

##### 8.9.1.1

##### Concept of a Weight Vector

Consider the simple two class multispectral space shown in Fig. 8.11, which has been constructed intentionally so that a simple straight line can be drawn between the pixels as shown. This straight line, which will be a multidimensional linear surface in general and which is called a hyperplane, can function as a decision surface for classification. In the two dimensions shown, the equation of the line can be expressed

$$w_1x_1 + w_2x_2 + w_3 = 0$$

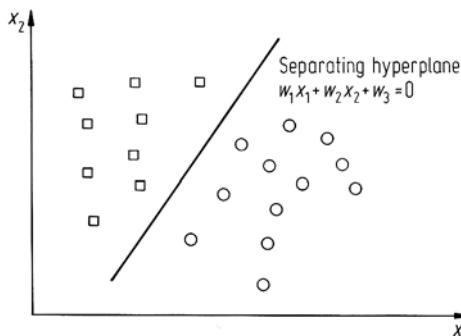
where the  $x_i$  are the brightness value co-ordinates of the multispectral space and the  $w_i$  are a set of coefficients, usually called weights. There will be as many weights as the number of channels in the data, plus one. In general, if the number of channels or bands is  $N$ , the equation of a linear surface is

$$w_1x_1 + w_2x_2 + \dots + w_Nx_N + w_{N+1} = 0$$

which can be written as

$$\mathbf{w}^T \mathbf{x} + w_{N+1} = 0 \quad (8.22)$$

where  $\mathbf{x}$  is the co-ordinate vector and  $\mathbf{w}$  is called the weight vector. The transpose operation has the effect of turning the column vector into a row vector so that the product gives the correct expanded form of the previous equation.



**Fig. 8.11.** Two dimensional multispectral space, with two classes of pixel that can be separated by a linear surface

In a real exercise the position of the separating surface would be unknown initially. Training a linear classifier amounts to determining an appropriate set of the weights that places the decision surface between the two sets of training samples. There is not necessarily a unique solution – any of an infinite number of (marginally different) decision hyperplanes will suffice to separate the two classes.

For a given data set, an explicit equation for the separating surface can be obtained using the minimum distance rule, as discussed in Sect. 8.3, which entails finding the mean vectors of the two class distributions. An alternative method is outlined in the following, based on selecting an arbitrary surface and then iterating it into an acceptable position. Even though not often used anymore, this method is useful to consider since it establishes some of the concepts used in neural networks and support vector machines (see Sect. 8.9.2).

### 8.9.1.2

#### Testing Class Membership

The calculation in (8.22) will be exactly zero only for values of  $\mathbf{x}$  lying on the decision surface. If we substitute into that equation values of  $\mathbf{x}$  corresponding to the pixel points indicated in Fig. 8.11 the left hand side will be non-zero. For pixels in one class a positive result will be given, while pixels on the other side will give a negative result. Thus, once the decision surface has been identified (i.e. trained), then a decision rule is

$$\begin{aligned} \mathbf{x} &\in \text{class 1 if } \mathbf{w}'\mathbf{x} + w_{N+1} > 0 \\ \mathbf{x} &\in \text{class 2 if } \mathbf{w}'\mathbf{x} + w_{N+1} < 0 \end{aligned} \quad (8.23)$$

### 8.9.1.3

#### Training

A full discussion of linear classifier training is given in Nilsson (1965, 1990); only those aspects helpful to the neural network development following are treated here.

It is expedient to define a new, augmented pixel vector according to

$$\mathbf{y} = [\mathbf{x}', 1]^t$$

If, in (8.22), we also take the term  $w_{N+1}$  into the definition of the weight vector, viz.

$$\mathbf{w} = [\mathbf{w}', w_{N+1}]^t$$

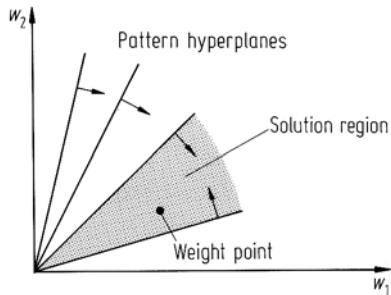
then the equation of the decision surface, can be expressed more compactly as

$$\mathbf{w}'\mathbf{y} = 0 \quad (\text{or equivalently } \mathbf{w} \cdot \mathbf{y} = 0)$$

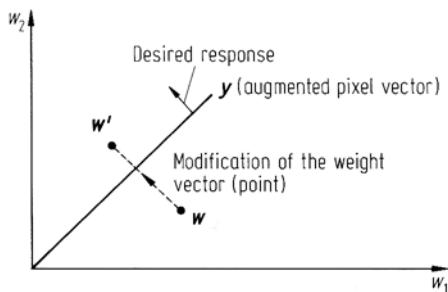
so that the decision rule of (8.23) can be restated

$$\begin{aligned} \mathbf{x} &\in \text{class 1 if } \mathbf{w}'\mathbf{y} > 0 \\ \mathbf{x} &\in \text{class 2 if } \mathbf{w}'\mathbf{y} < 0 \end{aligned} \quad (8.24)$$

We usually think of  $\mathbf{w}'\mathbf{y} = 0$  as defining a linear surface in the  $\mathbf{x}$  (or now  $\mathbf{y}$ ) multispectral space, in which the coefficients of the variables ( $y_1, y_2$ , etc.) are the



**Fig. 8.12.** Representation of pixels as hyperplanes and weight vectors as points in so-called weight space. The arrows indicate the side of each pixel plane on which the weight point must lie for correct classification



**Fig. 8.13.** Modification of the weight point to give the correct response

weights  $w_1$ ,  $w_2$ , etc. However it is also possible to think of the equation as describing a linear surface in which the  $y$ 's are the coefficients and the  $w$ 's are the variables. This interpretation will see these surfaces plotted in a co-ordinate system which has axes  $w_1$ ,  $w_2$ , etc. A two-dimensional version of this *weight space*, as it is called, is shown in Fig. 8.12, in which have been plotted a number of *pattern hyperplanes*; these are specific linear surfaces in the new co-ordinates that pass through the origin and have, as their coefficients, the components of the (augmented) pixel vectors. Thus, while the pixels plot as points in multispectral space, they plot as linear surfaces in weight space. Likewise, a set of weight coefficients will define a surface in multispectral space, but will plot as a point in weight space. Although this is an abstract concept it will serve to facilitate an understanding of how a linear classifier can be trained.

In weight space the decision rule of (8.24) still applies – however now it tests that the *weight point* is on the appropriate side of the *pattern hyperplane*. For example, Fig. 8.12 shows a single weight point which lies on the correct side of each pixel and thus defines a suitable decision surface in multispectral space. In the diagram, small arrows are attached to each pixel hyperplane to indicate the side on which the weight point must lie in order that the test of (8.24) succeeds for all pixels. The purpose of training the linear classifier is to ensure that the weight point is located somewhere within the *solution region*. If, through some initial guess, the weight point is located somewhere else in weight space then it has to be moved to the solution region.

Suppose an initial guess is made for the weight vector  $w$ , but that this places the weight point on the wrong side of a particular pixel hyperplane as illustrated in Fig. 8.13. Clearly, the weight point has to be shifted to the other side to give a

correct response in (8.24). The most direct manner in which the weight point can be modified is to move it straight across the pixel hyperplane. This can be achieved by adding a scaled amount of the pixel vector to the weight vector<sup>4</sup>. The new position of the weight point is then

$$\mathbf{w}' = \mathbf{w} + c\mathbf{y} \quad (8.25)$$

where  $c$  is called the correction increment, the size of which determines by how much the original weight point is moved orthogonal to the pixel hyperplane. If it is large enough the weight point will be shifted right across the pixel plane, as required. Having so modified the weight vector, the product in (8.24) then becomes

$$\begin{aligned}\mathbf{w}'^t \mathbf{y} &= \mathbf{w}^t \mathbf{y} + c\mathbf{y}^t \mathbf{y} \\ &= \mathbf{w}^t \mathbf{y} + c|\mathbf{y}|^2\end{aligned}$$

Clearly, if the initial  $\mathbf{w}^t \mathbf{y}$  was erroneously negative a suitable positive value of  $c$  will give a positive value of  $\mathbf{w}'^t \mathbf{y}$ ; otherwise a negative value of  $c$  will correct an erroneous initial positive value of the product.

Using the class membership test in (8.24) and the correction formula of (8.25) the following iterative nonparametric training procedure, referred to as *error correction feedback*, is adopted.

First, an initial position for the weight point is chosen arbitrarily. Then, pixel vectors from training sets are presented one at a time. If the current weight point position classifies a pixel correctly then no action need be taken; otherwise the weight vector is modified as in (8.25) with respect to that particular pixel vector. This procedure is repeated for each pixel in the training set, and the set is scanned as many times as necessary to move the weight point into the solution region. If the classes are linearly separable then such a solution will be found.

### 8.9.1.4 Setting the Correction Increment

Several approaches can be adopted for choosing the value of the correction increment,  $c$ . The simplest is to set  $c$  equal to a positive or negative constant (according to the change required in the  $\mathbf{w}^t \mathbf{y}$  product). A common choice is to make  $c = \pm 1$  so that application of (8.25) amounts simply to adding the augmented pixel vector to or subtracting it from the weight vector, thereby obviating multiplications and giving fast training.

Another rule is to choose the correction increment proportional to the difference between the desired and actual response of the classifier:

$$c = \eta(t - \mathbf{w}^t \mathbf{y})$$

---

<sup>4</sup> The hyperplane in  $\mathbf{w}$  coordinates is given by  $\mathbf{w}^t \mathbf{y} = 0$ ; a vector normal to that hyperplane is the vector of the coefficients of  $\mathbf{w}$ . This can be checked for a simple two dimensional example. A line through the origin with unity slope is  $-w_1 + w_2 = 0$ . A vector normal to the line joins the origin to  $(-1, 1)$ , i.e.  $\mathbf{y} = [-1, 1]^t$ .

so that (8.25) can be written

$$\mathbf{w}' = \mathbf{w} + \Delta \mathbf{w}$$

with

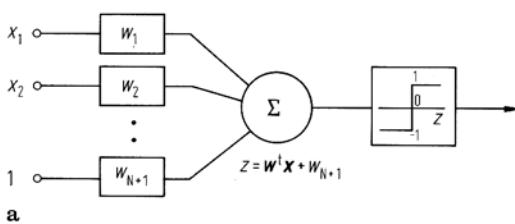
$$\Delta \mathbf{w} = \eta(t - \mathbf{w}'^T \mathbf{y}) \mathbf{y} \quad (8.26)$$

where  $t$  is the desired response to the training pattern  $\mathbf{y}$  and  $\mathbf{w}'^T \mathbf{y}$  is the actual response;  $\eta$  is a factor which controls the degree of correction applied. Usually  $t$  would be chosen as +1 for one class and -1 for the other.

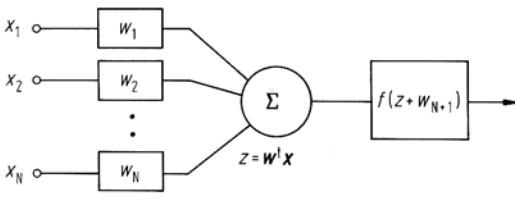
### 8.9.1.5

#### Classification – The Threshold Logic Unit

After the linear, two category classifier has been trained, so that the final version of the weight vector  $\mathbf{w}$  is available, it is ready to be presented with pixels it has not seen before in order to attach ground cover class labels to those pixels. This is achieved through application of the decision rule in (8.24). It is useful, in anticipation of neural networks, to picture the classification rule in diagrammatic form as depicted in Fig. 8.14a. Simply, this consists of weighting elements, a summing device and an output element which, in this case, performs the maximum selection. Together these are referred to as a *threshold logic unit* (TLU). It bears substantial similarity to the concept of a *processing element* used in neural networks for which the output thresholding unit is replaced by a more general function and the pathway for the unity input in the augmented pattern vector is actually incorporated into the output function. The latter can be done for a simple TLU as shown in Fig. 8.14b, in which the simple thresholding element has been replaced by a functional block which performs



a



b

**Fig. 8.14.** a Diagrammatic representation of (8.24). b More useful representation of a processing element in which the thresholding function is generalised

the addition of the final weighting coefficient to the weighted sum of the input pixel components, and then performs a thresholding (or more general nonlinear) operation.

### 8.9.1.6 Multicategory Classification

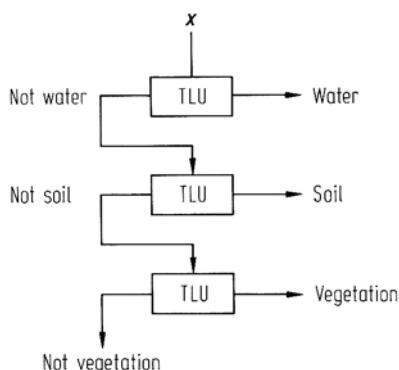
The foregoing work on linear classification has been based on an approach that can perform separation of pixel vectors into just two categories. Were it to be considered for remote sensing, it needs to be extended to be able to cope with a multiclass problem.

Multicategory classification can be carried out in one of two ways. First a decision tree of linear classifiers (TLUs) can be constructed, as seen in Fig. 8.15, at each decision node of which a decision of the type (water or not water) is made. At a subsequent node the (not water) category might be differentiated as (soil or not soil) etc. It should be noted that the decision process at each node has to be trained separately.

Alternatively, a multicategory version of the simple binary linear classifier can be derived. This reverts, for its derivation, to the concept of a discriminant function and, specifically, defines the linear classifier discriminant function for class  $i$  as

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{y} \quad i = 1, \dots, M$$

Class membership is then decided on the basis of the usual decision rule expressed in (8.11a), i.e. according to the largest of the  $g_i(\mathbf{x})$  for the given pixel vector  $\mathbf{x}$ . For training, an initial arbitrary set of weight vectors and thus discriminant functions is chosen. Then each of the training pixels is checked in turn. Suppose, for a particular pixel the  $j$ th discriminant function is erroneously largest, when in fact the pixel belongs the  $i$ th category. A correction is carried out by adjusting the weight vectors for these two discriminant functions, to increase that for the correct class for the pixel and to decrease that for the incorrect class, according to



**Fig. 8.15.** Binary decision tree of TLUs used for multicategory classification

$$\begin{aligned}\mathbf{w}'_i &= \mathbf{w}_i + c\mathbf{y} \\ \mathbf{w}'_j &= \mathbf{w}_j - c\mathbf{y}\end{aligned}\tag{8.27}$$

where  $c$  is the correction increment. Again this correction procedure is iterated over the training set of pixels as many times as necessary to obtain a solution. Nilsson (1965, 1990) shows that a solution is possible by this approach.

### 8.9.2

#### Support Vector Classifiers

##### 8.9.2.1

###### Linearly Separable Data

The training process outlined in Sect. 8.9.1.3 can lead to many, non-unique, yet acceptable solutions for the weight vector. The actual size of the solution region depicted in Fig. 8.12 is an indication of that. Also, every training pixel takes part in the training process; yet examination of Fig. 8.11 suggests that it is only those pixels in the vicinity of the separating hyperplane that define where the hyperplane needs to lie in order to give a reliable classification.

The support vector machine (SVM) provides a training approach that depends only on those pixels in the vicinity of the separating hyperplane (called the support *pixel* vectors). It also leads to a hyperplane position that is in a sense optimal for the available training patterns, as will be seen shortly.

The support vector concept was introduced to remote sensing image classification by Gualtieri and Crompt (1998). Two recent reviews that contain more detail than is given in the following treatment are by Burges (1998) and Huang et al. (2002). A very good recent treatment from a remote sensing perspective has been given by Melgani and Bruzzone (2004).

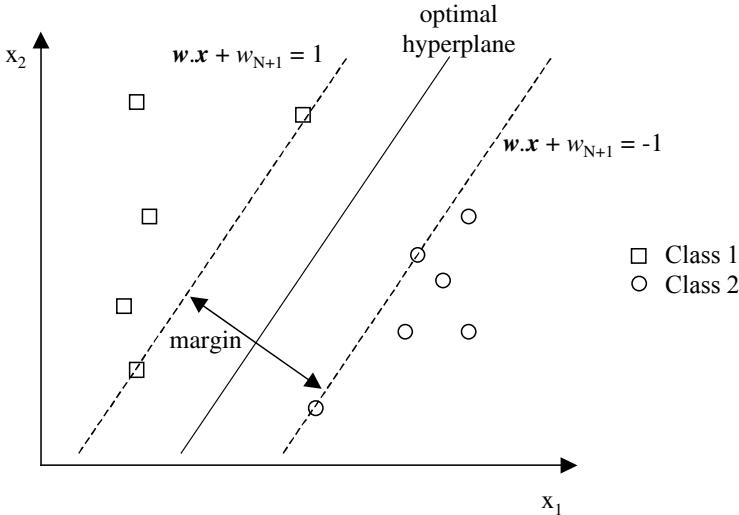
If we expand the region in the vicinity of the hyperplane in Fig. 8.11 we can see, as suggested in Fig. 8.16, that the optimal orientation of the hyperplane is when there is a maximum separation between the patterns in the two classes. We can then draw two further hyperplanes parallel to the separating hyperplane, as shown, bordering the nearest training pixels from the two classes. The equations for the hyperplanes are shown in the figure. Note that the choice of unity on the right hand side of the equations for the two marginal hyperplanes is arbitrary, but it helps in the analysis. If it were otherwise it could be scaled to unity by appropriately scaling the weighting coefficients  $w_k$ . Note that for pixels that lie *beyond the marginal hyperplanes*, we have

$$\text{for class 1 pixels} \quad \mathbf{w} \cdot \mathbf{x} + w_{N+1} \geq 1 \tag{8.28a}$$

$$\text{for class 2 pixels} \quad \mathbf{w} \cdot \mathbf{x} + w_{N+1} \leq -1 \tag{8.28b}$$

It is useful now to describe the class label of the  $i$ th pixel by the variable  $y_i$ , which takes the value +1 for class 1 and -1 for class 2 pixels. Equations (8.28a) and (8.28b) can then be written as a single expression valid for pixels from both classes:

$$(\mathbf{w} \cdot \mathbf{x} + w_{N+1})y_i \geq 1 \text{ for pixel } i \text{ in its correct class.}$$



**Fig. 8.16.** Expanded version of Fig. 8.11 showing that an optimal separating hyperplane orientation exists determined by finding the maximum separation between the training pixels. Under that condition two marginal hyperplanes can be constructed using only those pixels vectors closest to the separating surface

Alternatively

$$w \cdot x + w_{N+1}) y_i - 1 \geq 0 \quad (8.29)$$

Equation (8.29) must hold for all pixels if the data is linearly separated by the two marginal hyperplanes of Fig 8.16. Those hyperplanes, defined by the equalities in (8.28), are described by

$$w \cdot x + w_{N+1} - 1 = 0$$

$$w \cdot x + w_{N+1} + 1 = 0$$

The perpendicular distances of these hyperplanes from the origin, respectively, are  $-(w_{N+1} - 1)/\|w\|$  and  $-(w_{N+1} + 1)/\|w\|$ , where  $\|w\|$  is the Euclidean length of the weight vector. Therefore, the distance between the two hyperplanes, which is the *margin* in Fig. 8.16, is  $2/\|w\|$ .

The best position (orientation) for the separating hyperplane will be that for which  $2/\|w\|$  is a maximum, or equivalently when the magnitude of the weight vector,  $\|w\|$ , is a minimum. However there is a constraint! As we seek to maximise the margin between the two marginal hyperplanes by minimising  $\|w\|$  we must not allow (8.29) to be invalidated. In other words, all the training pixels must be on their correct side of the marginal hyperplanes. We handle the process of minimising  $\|w\|$  subject to that constraint by the process known as Lagrange multipliers. This requires us to set up a function (called the Lagrangian) which includes the expression to be minimised ( $\|w\|$ ) from which is subtracted a proportion ( $\alpha_i$ ) of each constraint (one for each training pixel) in the following manner:

$$L = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i \{y_i(\mathbf{w} \cdot \mathbf{x}_i + w_{N+1}) - 1\} \quad (8.30)$$

The  $\alpha_i$  are called Lagrange multipliers and are positive by definition, i.e.

$$\alpha_i \geq 0 \text{ for all } i.$$

By minimising  $L$  we minimise  $\|\mathbf{w}\|$  subject to the constraint (8.29).

In (8.30) it is convenient to substitute

$$f(\mathbf{x}_i) = (\mathbf{w} \cdot \mathbf{x}_i + w_{N+1})y_i - 1$$

to give

$$L = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_i \alpha_i f(x_i)$$

noting that for pixels in their correct class  $f(\mathbf{x}_i) \geq 0$ .

It is useful here to remember what our task is. We have to find the *most* separated marginal hyperplanes in Fig. 8.16. In other words we need to find the  $\mathbf{w}$  and  $w_{N+1}$  that minimises  $L$  and thus maximises the *margin* shown in the figure.

But in seeking to minimise  $L$  (essentially during the training process) how do we treat the  $\alpha_i$ ? Suppose (8.29) is violated, as could happen for some pixels during training; then  $f(\mathbf{x}_i)$  will be negative. Noting that  $\alpha_i$  is positive that would cause  $L$  to increase. But we need to find values for  $\mathbf{w}$  and  $w_{N+1}$  such that  $L$  is minimised. The worst possible case to handle is when the  $\alpha_i$  are such as to cause  $L$  to be a maximum, since that forces us to minimise  $L$  with respect to  $\mathbf{w}$  and  $w_{N+1}$  while the  $\alpha_i$  are trying to make it as large as possible. The most robust approach to finding  $\mathbf{w}$  and  $w_{N+1}$  (and thus the hyperplanes) therefore is to find the values of  $\mathbf{w}$  and  $w_{N+1}$  that minimise  $L$  while simultaneously finding the  $\alpha_i$  that try to maximise it.

Thus we require, first, that:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_i \alpha_i y_i \mathbf{x}_i = 0$$

$$\text{so that } \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \quad (8.31a)$$

Secondly we require:

$$\frac{\partial L}{\partial w_{N+1}} = - \sum_i \alpha_i y_i = 0$$

$$\text{so that } \sum_i \alpha_i y_i = 0. \quad (8.31b)$$

Before proceeding, examine (8.30) again, this time for training pixels that satisfy the requirement of (8.29). What value(s) of  $\alpha_i$  in (8.30) for those pixels maximise  $L$ ? Since  $y_i(\mathbf{w} \cdot \mathbf{x}_i + w_{N+1}) - 1$  is now always positive then the only (non-negative) value of  $\alpha_i$  that makes  $L$  as big as possible is  $\alpha_i = 0$ . Therefore, for any training

pixels on the correct side of the marginal hyperplanes,  $\alpha_i = 0$ . This is an amazing, yet intuitive, result. It says we do not have to use any of the training pixel vectors, other than those that reside exactly on one of the marginal hyperplanes. The latter are called *support vectors* since they are the only ones that support the process of finding the marginal hyperplanes. Thus, in applying (8.31a) to find  $\mathbf{w}$  we only have to use those pixels on the marginal hyperplanes.

But the training is not yet finished! We still have to find the relevant  $\alpha_i$  (i.e. those that maximise  $L$  and are non-zero).

To proceed, note that we can put  $\|\mathbf{w}\| = \mathbf{w} \cdot \mathbf{w}$  in (8.30). Now (8.30), along with (8.31a), can be written most generally as:

$$\begin{aligned} L &= \frac{1}{2} \left( \sum_i \alpha_i y_i \mathbf{x}_i \right) \cdot \left( \sum_j \alpha_j y_j \mathbf{x}_j \right) \\ &\quad - \sum_i \alpha_i \left[ y_i \left( \left( \sum_j \alpha_j y_j \mathbf{x}_j \right) \cdot \mathbf{x}_i + w_{N+1} \right) - 1 \right] \\ &= \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j - w_{N+1} \sum_i \alpha_i y_i + \sum_i \alpha_i \end{aligned}$$

Using (8.31b) this simplifies to

$$L = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (8.32)$$

which has to be maximised by the choice of  $\alpha_i$ . This usually requires a numerical procedure to solve for any real problem. Once we have found the  $\alpha_i$  – call them  $\alpha_i^o$  – we can substitute them into (8.31a) to give the optimal training vector:

$$\mathbf{w}^o = \sum_i \alpha_i^o y_i \mathbf{x}_i \quad (8.33a)$$

But we still do not have a value for  $w_{N+1}$ . Recall that on a marginal hyperplane

$$(\mathbf{w} \cdot \mathbf{x}_i + w_{N+1}) y_i - 1 = 0 .$$

Choose two support (training) vectors  $\mathbf{x}(1)$  and  $\mathbf{x}(-1)$  on each of the two marginal hyperplanes respectively for which  $y = 1$  and  $-1$ . For these vectors we have

$$\mathbf{w} \cdot \mathbf{x}(1) + w_{N+1} - 1 = 0$$

and

$$-\mathbf{w} \cdot \mathbf{x}(-1) - w_{N+1} - 1 = 0$$

so that

$$w_{N+1} = \frac{1}{2} (\mathbf{w} \cdot \mathbf{x}(1) + \mathbf{w} \cdot \mathbf{x}(-1)) . \quad (8.33b)$$

Normally sets of  $\mathbf{x}(1)$ ,  $\mathbf{x}(-1)$ , would be used, with  $w_{N+1}$  found by averaging.

With the values of  $\alpha_i^\circ$  determined by numerical optimisation, (8.33a,b) now give the parameters of the separating hyperplane that provides the largest margin between the two sets of training data. In terms of the training data, the equation of the hyperplane is:

$$\mathbf{w}^\circ \cdot \mathbf{x} + w_{N+1} = 0$$

so that the discriminant function, for an unknown pixel  $\mathbf{x}$  is

$$g(\mathbf{x}) = \text{sgn} (\mathbf{w}^\circ \cdot \mathbf{x} + w_{N+1}) \quad (8.34)$$

### 8.9.2.2

#### Linear Inseparability – The Use of Kernel Functions

If the pixel space is not linearly separable then the development of the previous section will not work without modification. A transformation of the pixel vector  $\mathbf{x}$  to a different (usually higher order) feature space can be applied that renders the data linearly separable allowing the earlier material to be applied.

The two significant equations for the linear support vector approach of the preceding section are (8.32) (for finding  $\alpha_i^\circ$ ) and (8.34) (the resulting discriminant function). By using (8.33a), (8.34) can be rewritten

$$g(\mathbf{x}) = \text{sgn} \left\{ \sum \alpha_i^\circ y_i \mathbf{x}_i \cdot \mathbf{x} + w_{N+1} \right\} \quad (8.35)$$

Now introduce the feature space transformation  $\mathbf{x} \rightarrow \Phi(\mathbf{x})$  so that (8.32) and (8.35) become

$$L = \sum_i \alpha_i - 0.5 \sum_{i,j} \alpha_i \alpha_j y_i y_j \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \quad (8.36a)$$

$$g(\mathbf{x}) = \text{sgn} \left\{ \sum \alpha_i^\circ y_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + w_{N+1} \right\} \quad (8.36b)$$

In both (8.32) and (8.35) the pixel vectors occur only in the dot products. As a result the  $\Phi(\mathbf{x})$  also appear only in dot products. So to use (8.36) it is strictly not necessary to know  $\Phi(\mathbf{x})$  but only a scalar quantity equivalent to  $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ .

We call the product  $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$  a *kernel*, and represent it by  $k(\mathbf{x}_i, \mathbf{x}_j)$  so that (8.36) becomes

$$L = \sum_i \alpha_i - 0.5 \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$g(\mathbf{x}) = \text{sgn} \left\{ \sum \alpha_i^\circ y_i k(\mathbf{x}_i, \mathbf{x}) + w_{N+1} \right\}$$

Provided we know the form of the kernel we never actually need to know the underlying transformation  $\Phi(\mathbf{x})$ ! Thus, after choosing  $k(\mathbf{x}_i, \mathbf{x}_j)$  we then find the  $\alpha_i^\circ$  that maximise  $L$  and use that value in  $g(\mathbf{x})$  to perform a classification.

Two commonly used kernels in remote sensing are:

$$\text{The polynomial kernel } k(\mathbf{x}_i, \mathbf{x}_j) = [(\mathbf{x}_i) \cdot (\mathbf{x}_j) + 1]^p$$

$$\text{The radial basis function kernel } k(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}$$

in which  $p$  and  $\gamma$  are constants to be chosen.

### 8.9.2.3 Multicategory Classification

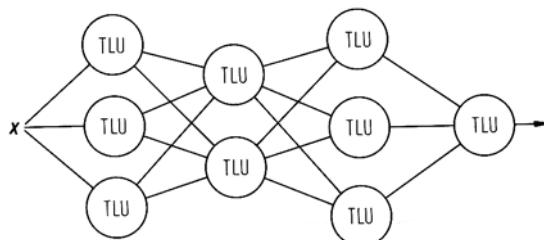
As in Sect. 8.9.1.6, the binary classifiers developed above can be used to perform multicategory classification by embedding them in a binary decision tree.

#### 8.9.3

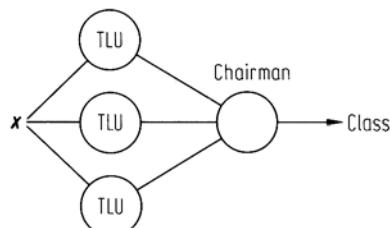
#### Networks of Classifiers – Solutions of Nonlinear Problems

The decision tree structure shown in Fig. 8.15 is a classifier network in that a collection of simple classifiers (in that case TLUs) is brought together to solve a complex problem. Nilsson (1965, 1990) has proposed a general network structure under the name of *layered classifiers* consisting entirely of interconnected TLUs, as shown in Fig. 8.17. The benefit of forming a classifier network is that data sets that are inherently not separable with a simple linear decision surface should, in principle, be able to be handled since the layered classifier is known to be capable of implementing nonlinear surfaces. The drawback however, is that training procedures for layered classifiers, consisting of TLUs, are difficult to determine.

One specific manifestation of a layered classifier, known as a committee machine, is depicted in Fig. 8.18. Here the first layer consists simply of a set of TLUs, to



**Fig. 8.17.** Layered TLU classifier



**Fig. 8.18.** Committee classifier

each of which a pixel vector under test is submitted to see which of two classes is recommended. The second layer is a single element which has the responsibility of judging the recommendations of each of the nodes in the first layer. It is therefore of the nature of a chairman or vote taker. It can make its decision on the basis of several sets of logic. First, it can decide class membership on the basis of the majority vote of the first layer recommendations. Secondly, it can decide on the basis of veto, in which all first layer classifiers have to agree before the vote taker will recommend a class. Thirdly, it could use a form of seniority logic in which the chairman rank orders the decisions of the first layer nodes. It always refers to one first. If that node has a solution then the vote taker accepts it and goes no further. Otherwise it consults the next most senior of the first layer nodes, etc. A committee classifier based on seniority logic has been developed for remote sensing applications by Lee and Richards (1985).

#### 8.9.4

#### The Neural Network Approach

For the purposes of this treatment a neural network is taken to be of the nature of a layered classifier such as depicted in Fig. 8.17, but with the very important difference that the nodes are not TLUs, although resembling them closely. The node structure in Fig. 8.14b can be made much more powerful, and coincidentally lead to a training theorem for multiclass nonlinear classification, if the output processing element does not apply a thresholding operation to the weighted input but rather applies a softer, and mathematically differentiable, operation.

##### 8.9.4.1

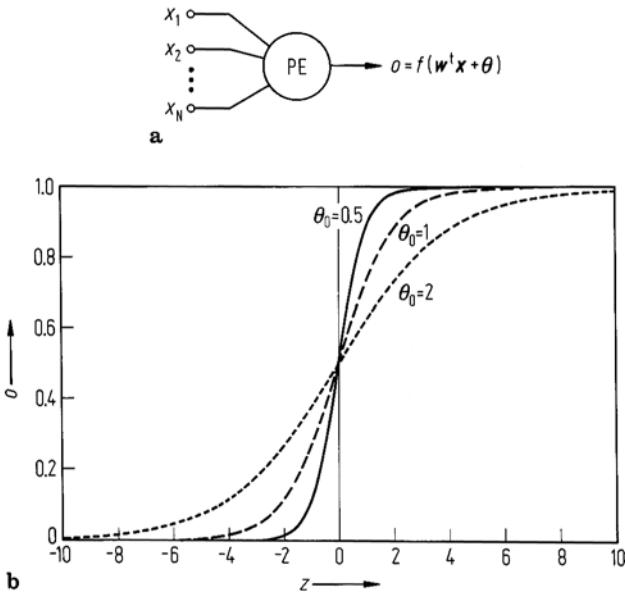
##### The Processing Element

The essential processing node in the neural network to be considered here (sometimes called a neuron by analogy to biological data processing from which the term neural network derives) is an element as shown in Fig. 8.14b with many inputs and with a single output, depicted simply in Fig. 8.19a. Its operation is described by

$$o = f(\mathbf{w}^t \mathbf{x} + \theta) \quad (8.37)$$

where  $\theta$  is a threshold (sometimes set to zero),  $\mathbf{w}$  is a vector of weighting coefficients and  $\mathbf{x}$  is the vector of inputs. For the special case when the inputs are the band values of a particular multispectral pixel vector it could be envisaged that the threshold  $\theta$  takes the place of the weighting coefficient  $w_{N+1}$  in (8.22). If the function  $f$  is a thresholding operation this processing element would behave as a TLU. In general, the number of inputs to a node will be defined by network topology as well as data dimensionality, as will become evident.

The major difference between the layered classifier of TLUs shown in Fig. 8.17 and the neural network, known as the multilayer perceptron, is in the choice of the function  $f$ , called the *activation function*. Its specification is simply that it emulate



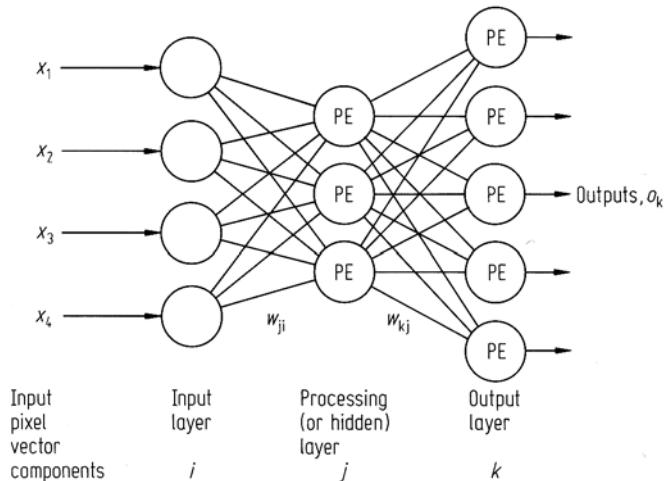
**Fig. 8.19.** a Neural network processing element. b Plots of (8.38) for various  $\theta_0$

thresholding in a soft or asymptotic sense and be differentiable. The most commonly encountered expression is

$$f(z) = \frac{1}{1 + e^{-z/\theta_0}} \quad (8.38)$$

where the argument  $z$  is  $w^T x + \theta$  as seen in (8.37) and  $\theta_0$  is a constant. This approaches 1 for  $z$  large and positive and 0 for  $z$  large and negative and is thus asymptotically thresholding. It is important to recognise that the outcome of the product  $w^T x$  is a simple scalar; when plotted with  $\theta = 0$ , (8.38) appears as shown in Fig. 8.19b. For  $\theta_0$  very small the activation function approaches a thresholding operation. Usually  $\theta_0 = 1$ .

A neural network for use in remote sensing image analysis will appear as shown in Fig. 8.20, being a layered classifier composed of processing elements of the type shown in Fig. 8.19a. It is conventionally drawn with an input layer of nodes (which has the function of distributing the inputs to the processing elements of the next layer, and scaling them if necessary) and an output layer from which the class labelling information is provided. In between there may be one or more so-called hidden or other processing layers of nodes. Usually one hidden layer will be sufficient, although the number of nodes to use in the hidden layer is often not readily determined. We return to this issue in Sect. 8.9.4.3 below.



**Fig. 8.20.** A multilayer perceptron neural network, and the nomenclature used in the derivation of the backpropagation training algorithm

### 8.9.4.2

#### Training the Neural Network – Backpropagation

Before it can perform a classification, the network of Fig. 8.20 must be trained. This amounts to using labelled training data to help determine the weight vector  $\mathbf{w}$  and the threshold  $\theta$  in (8.37) for each processing element connected into the network. Note that the constant  $\theta_0$  in (8.38), which governs the gradient of the activation function as seen in Fig. 8.19b, is generally pre-specified and does not need to be estimated from the training data.

Part of the complexity in understanding the training process for a neural net is caused by the need to keep careful track of the parameters and variables over all layers and processing elements, how they vary with the presentation of training pixels and (as it turns out) with iteration count. This can be achieved with a detailed subscript convention, or by the use of a simpler generalised notation. We will adopt the latter approach, following essentially the development given by Pao (1989). The derivation will be focussed on a 3 layer neural net, since this architecture has been found sufficient for many applications. However the results generalise to more layers.

Figure 8.20 incorporates the nomenclature used. The three layers are lettered as  $i$ ,  $j$ ,  $k$  with  $k$  being the output. The set of weights linking layer  $i$  PEs with those in layer  $j$  are represented generally by  $w_{ji}$ , while those linking layers  $j$  and  $k$  are represented by  $w_{kj}$ . There will be a very large number of these weights, but in deriving the training algorithm it is not necessary to refer to them all individually. Similarly the general activation function arguments  $z_i$  and outputs  $o_i$ , can be used to represent all the arguments and outputs in the corresponding layer.

For  $j$  and  $k$  layer PEs (8.37) is

$$o_j = f(z_j) \quad \text{with} \quad z_j = \sum_j w_{ji} o_i + \theta_j \quad (8.39a)$$

$$o_k = f(z_k) \quad \text{with} \quad z_k = \sum_k w_{kj} o_j + \theta_k \quad (8.39b)$$

The sums in (8.39) are shown with respect to the indices  $j$  and  $k$ . This should be read as meaning the sums are taken over all inputs of particular layer  $j$  and layer  $k$  PEs respectively. Note also that the sums are expressed in terms of the outputs of the previous layer since these outputs form the inputs to the PEs in question.

An untrained or poorly trained network will give erroneous outputs. Therefore, as a measure of how well a network is functioning during training, we can assess the outputs at the last layer ( $k$ ). A suitable measure along these lines is to use the sum of the squared output error. The error made by the network when presented with *a single training pixel* can thus be expressed

$$E = \frac{1}{2} \sum_k (t_k - o_k)^2 \quad (8.40)$$

where the  $t_k$  represent the desired or target outputs<sup>5</sup> and  $o_k$  represents the actual outputs from the output layer PEs in response to the training pixel. The factor of  $\frac{1}{2}$  is included for arithmetic convenience in the following. The sum is over all output layer PEs.

A useful training strategy is to adjust the weights in the processing elements until the error has been minimised, at which stage the actual outputs are as close as possible to the desired outputs.

A common approach for adjusting weights to reduce (and thus minimise) the value of a function of which they are arguments, is to modify their values proportional to the negative of the partial derivative of the function. This is called a gradient descent technique<sup>6</sup>. Thus for the weights linking the  $j$  and  $k$  layers let

$$w'_{kj} = w_{kj} + \Delta w_{kj}$$

with

$$\Delta w_{kj} = -\eta \frac{\partial E}{\partial w_{kj}}$$

where  $\eta$  is a positive constant that controls the amount of adjustment. This requires an expression for the partial derivative, which can be determined using the chain rule

---

<sup>5</sup> These will be specified from the training data labelling. The actual value taken by  $t_k$  however will depend on how the outputs themselves are used to represent classes. Each output could be a specific class indicator (e.g. 1 for class 1 and 0 class 2); alternatively some more complex coding of the outputs could be adopted. This is considered in Sect. 8.9.4.3.

<sup>6</sup> Another optimisation procedure used successfully for neural network training in remote sensing is the conjugate gradient method (Benediktsson et al., 1993).

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial o_k} \frac{\partial o_k}{\partial z_k} \frac{\partial z_k}{\partial w_{kj}} \quad (8.41)$$

each term of which must now be evaluated.

From (8.39b) and (8.38) we see (for  $\theta_0 = 1$ )

$$\frac{\partial o_k}{\partial z_k} = f'(z_k) = (1 - o_k)o_k \quad (8.42a)$$

and

$$\frac{\partial z_k}{\partial w_{kj}} = o_j \quad (8.42b)$$

Now from (8.40)

$$\frac{\partial E}{\partial o_k} = -(t_k - o_k) \quad (8.42c)$$

Thus the correction to be applied to the weights is

$$\Delta w_{kj} = \eta(t_k - o_k)(1 - o_k)o_k o_j \quad (8.43)$$

For a given trial, all of the terms in this expression are known so that a beneficial adjustments can be made to the weights which link the hidden layer to the output layer.

Now consider the weights that link the  $i$  and  $j$  layers. The weight adjustments are

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} = -\eta \frac{\partial E}{\partial o_j} \frac{\partial o_j}{\partial z_j} \frac{\partial z_j}{\partial w_{ji}}$$

In a similar manner to the above development we have

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial o_j} (1 - o_j)o_j o_i$$

Unlike the case with the output layer, however, we cannot obtain an expression for the remaining partial derivative from the error formula, since the  $o_j$  are not the outputs at the final layer, but rather those from the hidden layer. Instead we express the derivative in terms of a chain rule involving the output PEs. Specifically

$$\begin{aligned} \frac{\partial E}{\partial o_j} &= \sum_k \frac{\partial E}{\partial z_k} \frac{\partial z_k}{\partial o_j} \\ &= \sum_k \frac{\partial E}{\partial z_k} w_{kj} \end{aligned}$$

The remaining partial derivative can be obtained from (8.42a) and (8.42c) as

$$\frac{\partial E}{\partial z_k} = -(t_k - o_k)(1 - o_k)o_k$$

so that

$$\Delta w_{ji} = \eta(1 - o_j)o_j o_i \sum_k (t_k - o_k)(1 - o_k)o_k w_{kj} \quad (8.44)$$

Having determined the  $w_{kj}$  from (8.43), it is now possible to find values for the  $w_{ji}$  since all other entries in (8.44) are known or can be calculated readily.

For convenience we now define

$$\delta_k = (t_k - o_k)(1 - o_k)o_k \quad (8.45a)$$

and

$$\begin{aligned} \delta_j &= (1 - o_j)o_j \sum_k (t_k - o_k)(1 - o_k)o_k w_{kj} \\ &= (1 - o_j)o_j \sum_k \delta_k w_{kj} \end{aligned} \quad (8.45b)$$

so that we have

$$\Delta w_{kj} = \eta \delta_k o_j \quad (8.46a)$$

and

$$\Delta w_{ji} = \eta \delta_j o_i \quad (8.46b)$$

both of which should be compared with (8.26) to see the effect of a differentiable activation function.

The thresholds  $\theta_j$  and  $\theta_k$  in (8.39) are found in exactly the same manner as for the weights in that (8.46) is used, but with the corresponding inputs chosen to be unity.

Now that we have the mathematics in place it is possible to describe how training is carried out. The network is initialised with an arbitrary set of weights in order that it can function to provide an output. The training pixels are then presented one at a time to the network. For a given pixel the output of the network is computed using the network equations. Almost certainly the output will be incorrect to start with – i.e. the  $o_k$  will not match the desired class  $t_k$  for the pixel, as specified by its labelling in the training data. Correction to the output PE weights, described in (8.46a), is then carried out, using the definition of  $\delta_k$  in (8.45a). With these new values of  $\delta_k$  and thus  $w_{kj}$  (8.45b) and (8.46b) can be applied to find the new weight values in the earlier layers. In this way the effect of the output being in error is propagated back through the network in order to correct the weights. The technique is thus often referred to as *back propagation*.

Pao (1989) recommends that the weights not be corrected on each presentation of a single training pixel, but rather that the corrections for all pixels in the training set be aggregated into a single adjustment. Thus for  $p$  training patterns the bulk adjustments are<sup>7</sup>

$$\Delta w'_{kj} = \sum_p \Delta w_{kj} \quad \text{and} \quad \Delta w'_{ji} = \sum_k \Delta w_{ji}$$

---

<sup>7</sup> This is tantamount to deriving the algorithm with the error being calculated over all pixels  $p$  in the training set, viz  $E_p = \sum_p E$ , where  $E$  is the error expressed for a single pixel in (8.40).

After the weights have been so adjusted the training pixels are presented to the network again and the outputs re-calculated to see if they correspond better to the desired classes. Usually they will still be in error and the process of weight adjustment is repeated. Indeed the process is iterated as many times as necessary in order that the network respond with the correct class for each of the training pixels or until the number of errors in classifying the training pixels is reduced to an acceptable level.

#### **8.9.4.3**

#### **Choosing the Network Parameters**

When considering the use of the neural network approach to classification it is necessary to make several key decisions beforehand. First, the number of layers to use must be chosen. Generally, a three layer network is sufficient, with the purpose of the first layer being simply to distribute (or fan out) the components of the input pixel vector to each of the processing elements in the second layer. Thus the first layer does no processing as such, apart perhaps from scaling the input data, if required.

The next choice relates to the number of elements in each layer. The input layer will generally be given as many nodes as there are components (features) in the pixel vectors. The number to use in the output node will depend on how the outputs are used to represent the classes. The simplest method is to let each separate output signify a different class, in which case the number of output processing elements will be the same as the number of training classes. Alternatively, a single PE could be used to represent all classes, in which case a different value or level of the output variable will be attributed to each class. A further possibility is to use the outputs as a binary code, so that two output PEs can represent four classes, three can represent 8 classes and so on.

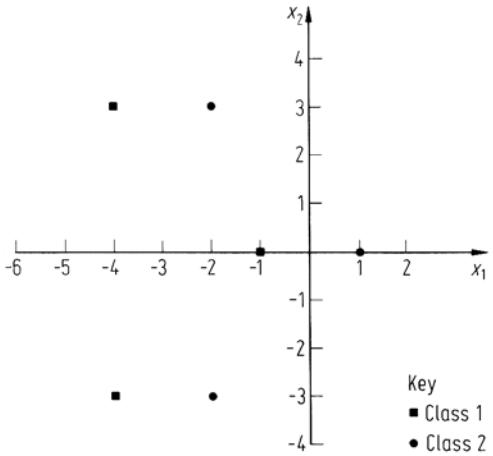
As a general guide the number of PEs to choose for the hidden or processing layers should be the same as or larger than the number of nodes in the input layer (Lippmann, 1987).

#### **8.9.4.4**

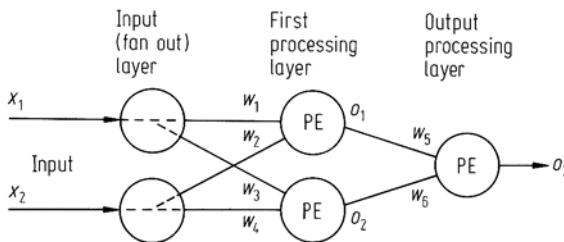
#### **Examples**

It is instructive to consider a simple example to see how a neural network is able to develop the solution to a classification problem. Figure 8.21 shows two classes of data, with three points in each, arranged so that they cannot be separated linearly. The network shown in Fig. 8.22 will be used to discriminate the data. The two PEs in the first processing layer are described by activation functions with no thresholds – i.e.  $\theta = 0$  in (8.37), while the single output PE has a non-zero threshold in its activation function.

Table 8.3 shows the results of training the network with the backpropagation method of the previous sections, along with the error measure of (8.40) at each step. It can be seen that the network approaches a solution quickly (approximately 50 iterations) but takes more iterations (approximately 250) to converge to a final result.



**Fig. 8.21.** Two-class data set, which is not linearly separable



$$o_1 = f(z_1), z_1 = w_1x_1 + w_2x_2$$

$$o_2 = f(z_2), z_2 = w_3x_1 + w_4x_2$$

$$o_3 = f(z_3), z_3 = w_5o_1 + w_6o_2 + \theta$$

**Fig. 8.22.** Two processing layer neural network to be applied to the data of Fig. 8.21

Having trained the network it is now possible to understand how it implements a solution to the nonlinear pattern recognition problem. The arguments of the activation functions of the PEs in the first processing layer each define a straight line (hyperplane in general) in the pattern space. Using the result at 250 iterations, these are:

$$2.901x_1 - 2.976x_2 = 0$$

$$2.902x_1 + 2.977x_2 = 0$$

which are shown plotted in Fig. 8.23. An individual line goes some way towards separating the data but cannot accomplish the task fully. It is now important to consider how the output PE operates on the outputs of the first layer PEs to complete the discrimination of the two classes. For pattern points lying exactly on one of the above lines, the output of the respective PE will be 0.5, given that the activation function of (8.38) has been used. However, for patterns a little distance away from

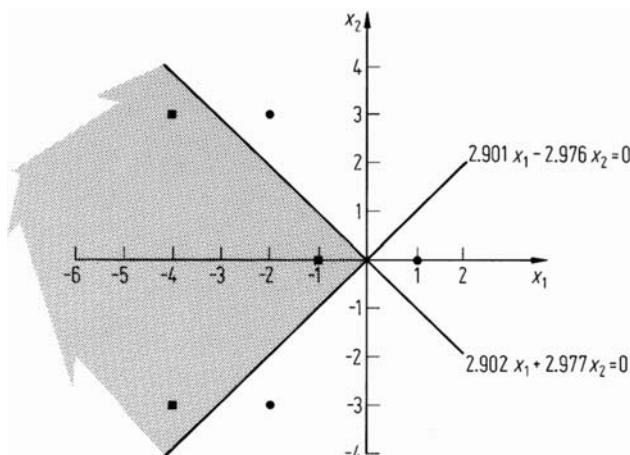
**Table 8.3.** Training the network of Fig. 8.22

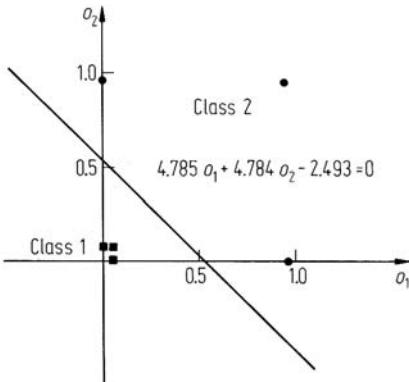
Iteration	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$	$w_6$	$\theta$	Error
0*	0.050	0.100	0.300	0.150	1.000	0.500	-0.500	0.461
1	0.375	0.051	0.418	0.121	0.951	0.520	-0.621	0.424
2	0.450	0.038	0.455	0.118	1.053	0.625	-0.518	0.408
3	0.528	0.025	0.504	0.113	1.119	0.690	-0.522	0.410
4	0.575	0.016	0.541	0.113	1.182	0.752	-0.528	0.395
5	0.606	0.007	0.570	0.117	1.240	0.909	-0.541	0.391
10	0.642	-0.072	0.641	0.196	1.464	1.034	-0.632	0.378
20	0.940	-0.811	0.950	0.882	1.841	1.500	-0.965	0.279
30	1.603	-1.572	1.571	1.576	2.413	2.235	-1.339	0.135
50	2.224	-2.215	2.213	2.216	3.302	3.259	-1.771	0.040
100	2.670	-2.676	2.670	2.677	4.198	4.192	-2.192	0.010
150	2.810	-2.834	2.810	2.835	4.529	4.527	-2.352	0.007
200	2.872	-2.919	2.872	2.920	4.693	4.692	-2.438	0.006
250	2.901	-2.976	2.902	2.977	4.785	4.784	-2.493	0.005

\* arbitrary initial set of weights and  $\theta$

**Table 8.4.** Response of the output layer PE

$o_1$	$o_2$	$o_3$
0	0	0.076 $\approx$ 0
0	1	0.908 $\approx$ 1
1	0	0.908 $\approx$ 1
1	1	0.999 $\approx$ 1

**Fig. 8.23.** Neural network solution for the data of Fig. 8.21



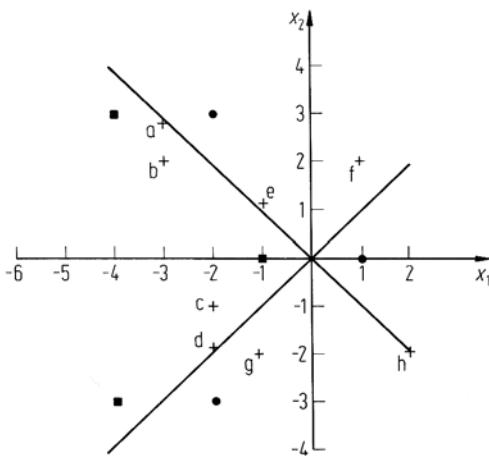
**Fig. 8.24.** Illustration of how the first processing layer PEs transform the input data into a linearly separable set, which is then discriminated by the output layer hyperplane

those lines the output of the first layer PEs will be close to 0 or 1 depending on which side of the hyperplane they lie. We can therefore regard the pattern space as being divided into two regions – 0 and 1 – by a particular hyperplane. Using these extreme values, Table 8.4 shows the possible responses of the output layer PE for patterns lying somewhere in the pattern space. As seen, for this example the output PE functions in the nature of a logical OR operation; patterns that lie on the 1 side of EITHER input PE hyperplane are labelled as belonging to one class, while those that lie on the 0 side of both hyperplanes will be labelled as belonging to the other class. Thus patterns which lie in the shaded region shown in Fig. 8.23 will generate a 0 at the output of the network and thus will be labelled as belonging to class 1, while patterns in the unshaded regions will generate a 1 response and thus will be labelled as belonging to class 2. Although this exercise is based only on two classes of data, similar functionality of the various PEs in a network can, in principle, be identified. The input PEs will always set up hyperplane divisions of the data and the later PEs will operate on the results of those simple discriminations.

An alternative way of considering how the network determines a solution is to regard the first processing layer PEs as transforming the data in such a way that later PEs (in this example only one) can exercise linear discrimination. Figure 8.24 shows a plot of the outputs of the first layer PEs when fed with the training data of Fig. 8.21. As observed, after transformation, the data is linearly separable. The hyperplane shown is that generated by the argument of the activation function of the output layer PE.

To illustrate how the network of Fig. 8.22 functions on unseen (i.e. testing set) data, Table 8.5 shows its response to the testing patterns indicated in Fig. 8.25. The class decision for a pattern is made by rounding the output PE response to 0 or 1 as appropriate. As noted, for this simple example, all patterns are correctly classified.

Benediktsson, Swain and Esroy (1990) have demonstrated the application of a neural network approach to classification in remote sensing, obtaining classification accuracies as high as 95% on training data although only as high as 52% when the network was applied to a test data set. It is suggested that the training data may not have been fully representative of the image. This is an important issue with neural



**Fig. 8.25.** Location of test data, indicated by the lettered crosses

**Table 8.5.** Performance of the network on the test data

Pattern	$x_1$	$x_2$	$z_1$	$o_1$	$z_2$	$o_2$	$z_3$	$o_3$	Class	
a	-3.0	2.8	-17.036	0.000	-	0.370	0.408	-0.539	0.368	1
b	-3.0	2.0	-14.655	0.000	-	2.752	0.056	-2.206	0.099	1
c	-2.0	-1.0	-	2.826	0.056	-	8.781	0.000	-2.224	0.098
d	-2.0	-1.94	-	0.029	0.493	-	11.579	0.000	-0.135	0.466
e	-1.0	1.1	-	6.175	0.002	-	0.373	0.592	0.350	0.587
f	1.0	2.0	-	3.051	0.045	-	8.856	1.000	2.506	0.925
g	-1.0	-2.0	-	3.051	0.955	-	8.856	0.000	2.077	0.889
h	2.0	-2.0	11.754	1.000	-	0.150	0.463	4.505	0.989	2

nets, more so than with statistical classification methods such as maximum likelihood, since the parameters in the statistical approach are *estimates* of statistics, and are not strongly affected by outlying training samples. The work of Benediktsson also illustrates, to an extent, the dependence of performance on the network architecture chosen.

Hepner (1990) has also used a neural network to perform a classification; in addition to the spectral properties of a pixel, however, he included the spectral measurements of the  $3 \times 3$  neighbourhood in order to allow spatial context to influence the labelling. Although quantitative accuracies are not given, Hepner is of the view that the results are better than when using a maximum likelihood classifier trained on spectral data only. Lippmann (1987) and Pao (1989) are good general references to consult for a wider treatment of neural network theory than has been given here, including other training methods. Both demonstrate also how neural networks can be used in unsupervised as well as supervised classification. Paola and Schowengerdt (1995a,b) provide a comprehensive review of the use of the multilayer Perception in remote sensing.

A range of neural network tools is available in MATLAB (1984–2004).

## References for Chapter 8

The classification techniques used in remote sensing image analysis come from the field of mathematical pattern recognition and, as a consequence, are covered extensively in the standard treatments in that discipline. Particular references that could be consulted to obtain more mathematical detail than has been provided above, and to see other variations on the algorithms often used, include Nilsson (1965, 1990), Duda, Hart and Stork (2001), and Tou and Gonzalez (1974). For each of these a strong mathematical background is required. The development of classification techniques in Swain and Davis (1978) is far less mathematical and includes introductory conceptual material on the probabilistic aspects of pattern recognition.

The classifiers treated in this chapter have all been single stage in that only one decision is made about a pixel as a result of which it is labelled as belonging to one of the available classes, or is left unclassified. Decision tree procedures are also possible. In these a series of decisions is taken in order to determine the correct label for a pixel. As an illustration, the first decision might allow a distinction to be made between water, shadow and fire burnt pixels on the one hand, and vegetation, urban and cleared regions on the other. Subsequent decisions then allow finer subdivisions leading ultimately to a single label for the pixel. Advantages of this approach include the fact that different sets of features can be used at each decision stage and indeed even different algorithms could be employed. Its use in remote sensing problems is described by Swain and Hauska (1977). The decision tree is one special example of the set of layered classification techniques (Nilsson, 1965, 1990). Section 11.8 treats decision trees.

Atkinson et al. (1985) discuss the value of image filtering prior to classification using both mean and median value methods. Demonstrations of probabilistic relaxation for pixel classification, and its comparison with simple maximum likelihood classification, will be found in Harris (1985), Gong and Howarth (1989) and Zenzo et al. (1987a, 1987b); the latter authors also introduce the concept of fuzzy relaxation. Development of the basic relaxation algorithm will be found in Rosenfeld, Hummel and Zucker (1976). Statistical methods for context classification have been developed by Swain et al. (1981), Kittler and Pairman (1985) and Khazenie and Crawford (1990). A selection of context classification methods are compared by Mohn et al. (1987).

For a first reading on neural networks, beyond the material presented here, the paper by Lippmann (1987) is recommended.

The seminal papers for the Markov Random Field approach for incorporating context in a labelling exercise are those by Geman and Geman (1984) and Besag (1986). One of the earliest treatments in remote sensing was that of Jeon and Landgrebe (1992), while Solberg et al (1996) have employed the method to incorporate spatial and temporal contexts. Jung and Swain (1996) have shown how the use of better estimates for class statistics, along with MRF for including spatial context, can lead to improved classification accuracy.

- P. Atkinson, J.L. Cushnie, J.R. Townshend and A. Wilson, 1985: Improving Thematic Mapper Land Cover Classification Using Filtered Data. *International Journal of Remote Sensing*, 6, 955–961.
- J.A. Benediktsson, P.H. Swain and O.K. Esroy, 1990: Neural Network Approaches Versus Statistical Methods in Classification of Multisource Remote Sensing Data. *IEEE Trans Geoscience and Remote Sensing*, 28, 540–552.
- J.A. Benediktsson, P.H. Swain and O.K. Esroy, 1993: Conjugate-Gradient Neural Networks in Classification of Multisource and Very-High-Dimensional Remote Sensing Data. *Int. J. Remote Sensing*, 14, 2883–2903.
- J. Besag, 1986: On the Statistical Analysis of Dirty Pictures. *J. Royal Statistical Soc. (B)*, 48, 259–302.

- C.T.C. Burges, 1998: A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121–167.
- B.V. Dasarathy, 1991: Nearest Neighbour (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, California.
- R.O. Duda, P.E. Hart and D.G. Stork, 2001: Pattern Classification, 2e, N.Y., Wiley.
- B.C. Forster, 1982: The Derivation of Approximate Equations to Correct for the Landsat MSS Point Spread Function. Proc. Commission 1 (Primary Data Acquisition) Int. Soc. for Photogrammetry and Remote Sensing, Canberra, April, 6–10.
- J.E. Freund, 1992: Mathematical Statistics, 5e, New Jersey, Prentice Hall.
- S. Geman and D. Geman, 1984: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Trans Pattern Analysis and Machine Intelligence*, PAMI-6, 721–740.
- P. Gong and P.J. Howarth, 1989: Performance Analyses of Probabilistic Relaxation Methods for Land-Cover Classification. *Remote Sensing of Environment*, 30, 33–42.
- P. Gong and P.J. Howarth, 1990: The Use of Structural Information for Improving Land-Cover Classification Accuracies at the Rural-Urban Fringe. *Photogrammetric Engineering and Remote Sensing*, 56, 67–73.
- J.A. Gualtieri and R.F. Cromp, 1999: Support Vector Machines for Hyperspectral Remote Sensing Classification. Proc. SPIE, 3584, 221–232.
- R. Harris, 1985: Contextual Classification Post-Processing of Landsat Data Using a Probabilistic Relaxation Model. *Int. J. Remote Sensing*, 6, 847–866.
- G.F. Hepner, 1990: Artificial Neural Network Classification Using a Minimal Training Set: Comparison to Conventional Supervised Classification. *Photogrammetric Engineering and Remote Sensing*, 56, 469–473.
- C. Huang, L.S. Davis and J.R.G. Townshend, 2002: An Assessment of Support Vector Machines for Land Cover Classification. *Int. J. Remote Sensing*, 23, 725–749.
- B. Jeon and D.A. Landgrebe, 1992: Classification with Spatio-Temporal Interpixel Class Dependency Contexts. *IEEE Trans. Geoscience and Remote Sensing*, 30, 663–672.
- Y. Jung and P.H. Swain, 1996: Bayesian Contextual Classification based on Modified M-estimates and Markov Random Fields. *IEEE Trans. Geoscience and Remote Sensing*, 34, 67–75.
- R.L. Kettig and D.A. Landgrebe, 1976: Classification of Multispectral Image Data by Extraction and Classification of Homogeneous Objects. *IEEE Trans. Geoscience Electronics*, GE-14, 19–26.
- N. Khazenie and M.M. Crawford, 1990: Spatial-Temporal Autocorrelation Model for Contextual Classification. *IEEE Trans. Geoscience and Remote Sensing*, 28, 529–539.
- J. Kittler and D. Pairman, 1985: Contextual Pattern Recognition Applied to Cloud Detection and Identification. *IEEE Trans Geoscience and Remote Sensing*, GE-23, 855–863.
- B.-C. Kuo and D.A. Landgrebe, 2002: A Robust Classification Procedure Based on Mixture Classifiers and Nonparametric Weighted Feature Extraction. *IEEE Trans. Geoscience and Remote Sensing*, 40, 2486–2494.
- T. Lee, 1984: Multisource Context Classification Methods in Remote Sensing. PhD Thesis, The University of New South Wales, Kensington, Australia.
- T. Lee and J.A. Richards, 1985: A Low Cost Classifier for Multitemporal Applications. *Int. J. Remote Sensing*, 6, 1405–1417.
- T. Lee and J.A. Richards, 1989: Pixel Relaxation Labelling Using a Diminishing Neighbourhood Effect. Proc. IGARSS'89. Vancouver, 634–637.
- R.P. Lippmann, 1987: An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, April, 4–22.

- MATLAB, 1984–2004: Neural Network Toolbox. The Math Works, Inc, MA.
- F. Melgani and L. Bruzzone, 2004: Classification of Hyperspectral Remote Sensing Images with Support Vector Machines. *IEEE Trans. Geoscience and Remote Sensing*, 42, 1778–1790.
- E. Mohn, N.L. Hjort and G.O. Storvik, 1987: A Simulation Study of Some Contextual Classification Methods for Remotely Sensed Data. *IEEE Trans. Geoscience and Remote Sensing*, 25, 796–804.
- N.J. Nilsson, 1965: Learning Machines. N.Y., McGraw-Hill.
- N.J. Nilsson, 1990: The Mathematical Foundations of Learning Machines. Palo Alto, Morgan Kaufmann.
- Y.H. Pao, 1989: Adaptive Pattern Recognition and Neural Networks. Reading, Addison-Wesley.
- J.D. Paola and R.A. Schowengerdt, 1995a: A Review and Analysis of Backpropagation Neural Networks for Classification of Remotely-Sensed Multi-Spectral Imagery. *Int. J. Remote Sensing*, 16, 3033–3058.
- J.D. Paola and R.A. Schowengerdt, 1995b: A Detailed Comparison of Backpropagation Neural Network and Maximum-Likelihood Classifiers for Urban Land Use Classification. *IEEE Trans. Geoscience and Remote Sensing*, 33, 981–996.
- S. Peleg and A. Rosenfeld, 1980: A New Probabilistic Relaxation Procedure. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-2, 362–369.
- J.A. Richards, D.A. Landgrebe and P.H. Swain, 1981: On the Accuracy of Pixel Relaxation Labelling. *IEEE Trans. Systems, Man and Cybernetics*, SMC-11, 303–309.
- A. Rosenfeld, R. Hummel and S. Zucker, 1976: Scene Labeling by Relaxation Algorithms. *IEEE Trans. Systems, Man and Cybernetics*, SMC-6, 420–433.
- B. Schölkopf and A. Smola, 2002: Learning with Kernels. Mass., MIT Press.
- A.H.S. Solberg, T. Taxt and A.K. Jain, 1996: A Markov Random Field Model for Classification of Multisource Satellite Imagery. *IEEE Trans. Geoscience and Remote Sensing*, 34, 100–113.
- P.H. Swain and S.M. Davis (Eds.), 1978: Remote Sensing: The Quantitative Approach, N.Y., McGraw-Hill.
- P.H. Swain and H. Hauska, 1977: The Decision Tree Classifier: Design and Potential. *IEEE Trans. Geoscience Electronics*, GE-15, 142–147.
- P.H. Swain, S.B. Varderman and J.C. Tilton, 1981: Contextual Classification of Multispectral Image Data. *Pattern Recognition*, 13, 429–441.
- J.T. Tou and R.C. Gonzalez, 1974: Pattern Recognition Principles, Mass., Addison-Wesley.
- F.E. Townsend, 1986: The Enhancement of Computer Classifications by Logical Smoothing. *Photogrammetric Engineering and Remote Sensing*, 52, 213–221.
- A.G. Wacker and D.A. Landgrebe, 1972: Minimum Distance Classification in Remote Sensing. First Canadian Symp. on Remote Sensing, Ottawa.
- S.D. Zenzo, R. Bernstein, S.D. Degloria and H.G. Kolsky, 1987: Gaussian Maximum Likelihood and Contextual Classification Algorithms for Multicrop Classification. *IEEE Trans. Geoscience and Remote Sensing*, 25, 805–814.
- S.D. Zenzo, S.D. Degloria, R. Bernstein and H.G. Kolsky, 1987: Gaussian Maximum Likelihood and Contextual Classification Algorithms for Multicrop Classification Experiments Using Thematic Mapper and Multispectral Scanner Sensor Data. *IEEE Trans. Geoscience and Remote Sensing*, 25, 815–824.

## Problems

**8.1** Suppose you have the following training data for three spectral classes, in which each pixel is characterised by only two spectral components  $\lambda_1$  and  $\lambda_2$ .

Class 1		Class 2		Class 3	
$\lambda_1$	$\lambda_2$	$\lambda_1$	$\lambda_2$	$\lambda_1$	$\lambda_2$
16	13	8	8	19	6
18	13	9	7	19	3
20	13	6	7	17	8
11	12	8	6	17	1
17	12	5	5	16	4
8	11	7	5	14	5
14	11	4	4	13	8
10	10	6	3	13	1
4	9	4	2	11	6
7	9	3	2	11	3

Develop the discriminant functions for a maximum likelihood classifier and use them to classify the patterns

$$\mathbf{x}_1 = \begin{bmatrix} 5 \\ 9 \end{bmatrix} \quad \mathbf{x}_2 = \begin{bmatrix} 9 \\ 8 \end{bmatrix} \quad \mathbf{x}_3 = \begin{bmatrix} 15 \\ 9 \end{bmatrix}$$

under the assumption of equal prior probabilities.

**8.2** Repeat question 1 but with the prior probabilities

$$\begin{aligned} p(1) &= 0.048 \\ p(2) &= 0.042 \\ p(3) &= 0.910 \end{aligned}$$

**8.3** Using the data of question 1 develop the discriminant functions for a minimum distance classifier and use them to classify the patterns  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$ .

**8.4** Develop a parallelepiped classifier from the training data given in question 1 and compare its classifications with those of the maximum likelihood classifier for the patterns  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  and  $\mathbf{x}_3$ , and the new pattern

$$\mathbf{x}_4 = \begin{bmatrix} 3 \\ 7 \end{bmatrix}$$

At the conclusion of the tests in questions 8.1, 8.3 and 8.4, it would be worthwhile sketching a multispectral (pattern) space and then locating in it the positions of the training data. Use this to form a subjective impression of the performance of each classifier in questions 8.1, 8.3 and 8.4.

**8.5** The following training data represents a subset of that in question 1 for just two of the classes. Develop discriminant functions for both maximum likelihood and minimum distance classifiers and use them to classify the patterns

$$\mathbf{x}_5 = \begin{bmatrix} 14 \\ 7 \end{bmatrix} \quad \mathbf{x}_6 = \begin{bmatrix} 20 \\ 13 \end{bmatrix}$$

Classify these patterns also using the minimum distance and maximum likelihood classifiers developed on the full training sets of question 1 and compare the results.

Class 1		Class 2	
$\lambda_1$	$\lambda_2$	$\lambda_1$	$\lambda_2$
11	12	17	8
10	10	16	4
14	11	14	5
		13	1

**8.6** Suppose a particular scene consists of just water and soil, and that a classification into these cover types is to be carried out on the basis of near infrared data using the maximum likelihood rule. When the thematic map is produced it is noticed that some water pixels are erroneously labelled as soil. How can this happen, and what steps could be taken to avoid it? Hint: Sketch some typical one dimensional normal distributions to represent the soil and water in infrared data, noting that soil would have a very large variance while that for water would be small. Remember the mathematical distribution functions extend to infinity.

# 9

## Clustering and Unsupervised Classification

### 9.1

#### Delineation of Spectral Classes

The successful application of maximum likelihood classification is dependent upon having delineated correctly the spectral classes in the image data of interest. This is necessary since each class is to be modelled by a normal probability distribution, as discussed in Chap. 8. If a class happens to be multimodal, and this is not resolved, then clearly the modelling cannot be very effective.

Users of remotely sensed data can only specify the information classes. Occasionally it might be possible to guess the number of spectral classes in a particular information class but, in general, the user would have little idea of the number of distinct unimodal groups that the data falls into in multispectral space. Gaussian mixture modelling can be used for this purpose (Sect. 8.7) but the complexity of estimating simultaneously the number of Gaussian components, and their parameters, can make this approach difficult to use. Clustering procedures are practical alternatives that can be used for that purpose; these are methods that have been applied in many data analysis fields to enable inherent data structures to be determined.

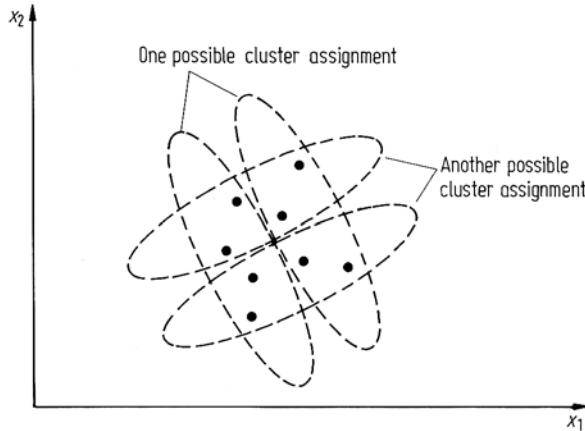
Clustering can also be used for unsupervised classification. In this technique an image is segmented into unknown classes. It is the task of the user to label those classes afterwards.

There are a great number of clustering methods. In this chapter only those commonly employed with remote sensing data are treated.

### 9.2

#### Similarity Metrics and Clustering Criteria

Clustering implies a grouping of pixels in multispectral space. Pixels belonging to a particular cluster are therefore spectrally similar. In order to quantify this relationship it is necessary to devise a similarity measure. Many similarity metrics have been proposed but those used commonly in clustering procedures are usually simple



**Fig. 9.1.** Two apparently acceptable clusterings of a set of two dimensional data

distance measures in multispectral space. The most frequently encountered are Euclidean distance and  $L_1$  (or interpoint) distance. If  $\mathbf{x}_1$  and  $\mathbf{x}_2$  are two pixels whose similarity is to be checked then the Euclidean distance between them is

$$\begin{aligned} d(\mathbf{x}_1, \mathbf{x}_2) &= ||\mathbf{x}_1 - \mathbf{x}_2|| \\ &= \{(\mathbf{x}_1 - \mathbf{x}_2)^t (\mathbf{x}_1 - \mathbf{x}_2)\}^{\frac{1}{2}} \\ &= \left\{ \sum_{i=1}^N (x_{1i} - x_{2i})^2 \right\}^{\frac{1}{2}} \end{aligned} \quad (9.1)$$

where  $N$  is the number of spectral components. The  $L_1$  distance between the pixels is

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i=1}^N |x_{1i} - x_{2i}|. \quad (9.2)$$

Clearly the latter is computationally faster to determine. However it can be seen as less accurate than the Euclidean distance measure.

By using a distance measure it should be possible to determine clusters in data. Often however there could be several acceptable clusters assignments of the data, as depicted in Fig. 9.1, so that once a candidate clustering has been found it is desirable to have a means by which the “quality” of clustering can be measured. The availability of such a measure should allow one cluster assignment of the data to be chosen over all others.

A common clustering criterion or quality indicator is the sum of squared error (SSE) measure, defined as

$$\begin{aligned} SSE &= \sum_{C_i} \sum_{\mathbf{x} \in C_i} (\mathbf{x} - \mathbf{m}_i)^t (\mathbf{x} - \mathbf{m}_i) \\ &= \sum_{C_i} \sum_{\mathbf{x} \in C_i} ||\mathbf{x} - \mathbf{m}_i||^2 \end{aligned} \quad (9.3)$$

where  $\mathbf{m}_i$  is the mean of the  $i$ th cluster and  $\mathbf{x} \in C_i$  is a pattern assigned to that cluster. The outer sum is over all the clusters. This measure computes the cumulative distance of each pattern from its cluster centre for each cluster individually, and then sums those measures over all the clusters. If it is small the distances from patterns to cluster means are all small and the clustering would be regarded favourably.

Other quality of clustering measures exist. One popular one is to derive a “within cluster scatter measure” by determining the average covariance matrix of the clusters, and a “between cluster scatter measure” by looking at the means of the clusters compared with the global mean of the data. These two measures are combined into a single figure of merit as discussed in Duda, Hart and Stork (2001) and Coleman and Andrews (1979). It can be shown that figures of merit such as these are similar to the sum of squared error criterion.

It is of interest to note that SSE has a theoretical minimum of zero, which corresponds to all clusters containing only a single data point. As a result, if an iterative method is used to seek the natural clusters or spectral classes in a set of data then it has a guaranteed termination point, at least in principle. In practice it may be too expensive to allow natural termination. Instead, iterative procedures are often stopped when an acceptable degree of clustering has been achieved.

It is possible now to consider the implementation of an actual clustering algorithm. While it should depend upon a progressive minimisation (and thus calculation) of SSE this is impracticable since it requires an enormous number of values of SSE for the evaluation of all candidate clusterings. For example, there are approximately  $C^P/C!$  ways of placing  $P$  patterns into  $C$  clusters (Duda, Hart and Stork, 2001). This number of SSE values would require computation at each stage of clustering to allow a minimum to be chosen. Rather than embark upon such a rigorous and computationally expensive approach the heuristic procedure of the following section is usually adopted in practice.

Similarity metrics for clustering can incorporate measures other than spectral likeness. Spatial proximity might be important in some applications as might components that account for categorical information. For example, clustering crop pixels might be guided by all of multispectral measurements, soil type and spatial contiguity. These more general metrics are not covered here.

### 9.3

## The Iterative Optimization (Migrating Means) Clustering Algorithm

The iterative optimization clustering procedure, also called the migrating means technique, is essentially the isodata algorithm presented by Ball and Hall (1965). It is based upon estimating some reasonable assignment of the pixel vectors into candidate clusters and then moving them from one cluster to another in such a way that the SSE measure of the preceding section is reduced.

### 9.3.1

#### The Basic Algorithm

The iterative optimization algorithm is implemented by the following set of basic steps:

1. The procedure is initialised by selecting  $C$  points in multispectral space to serve as candidate cluster centres. Let these be called

$$\hat{\mathbf{m}}_i, i = 1, \dots, C.$$

The selection of the  $\hat{\mathbf{m}}_i$  at this stage is arbitrary with the exception that no two may be the same. To avoid anomalous cluster generation with unusual data sets it is generally wise to space the initial cluster means uniformly over the data. This can also serve to enhance convergence.

Besides choosing the  $\hat{\mathbf{m}}_i$  the number of clusters  $C$ , must be specified beforehand by the user.

2. The location  $\mathbf{x}$  of each pixel in the segment of the image to be clustered is examined and the pixel is assigned to the nearest candidate cluster. This assignment would be made on the basis of the Euclidean or even  $L_1$  distance measure.
3. The new set of means that result from the grouping produced in Step 2 are computed. Let these be denoted

$$\mathbf{m}_i, i = 1, \dots, C.$$

4. If  $\mathbf{m}_i = \hat{\mathbf{m}}_i$  for all  $i$ , the procedure is terminated. Otherwise  $\hat{\mathbf{m}}_i$  is redefined as the current value of  $\mathbf{m}_i$  and the procedure returns to Step 2.

The iterative optimization procedure is illustrated for a simple set of two dimensional patterns in Fig. 9.2.

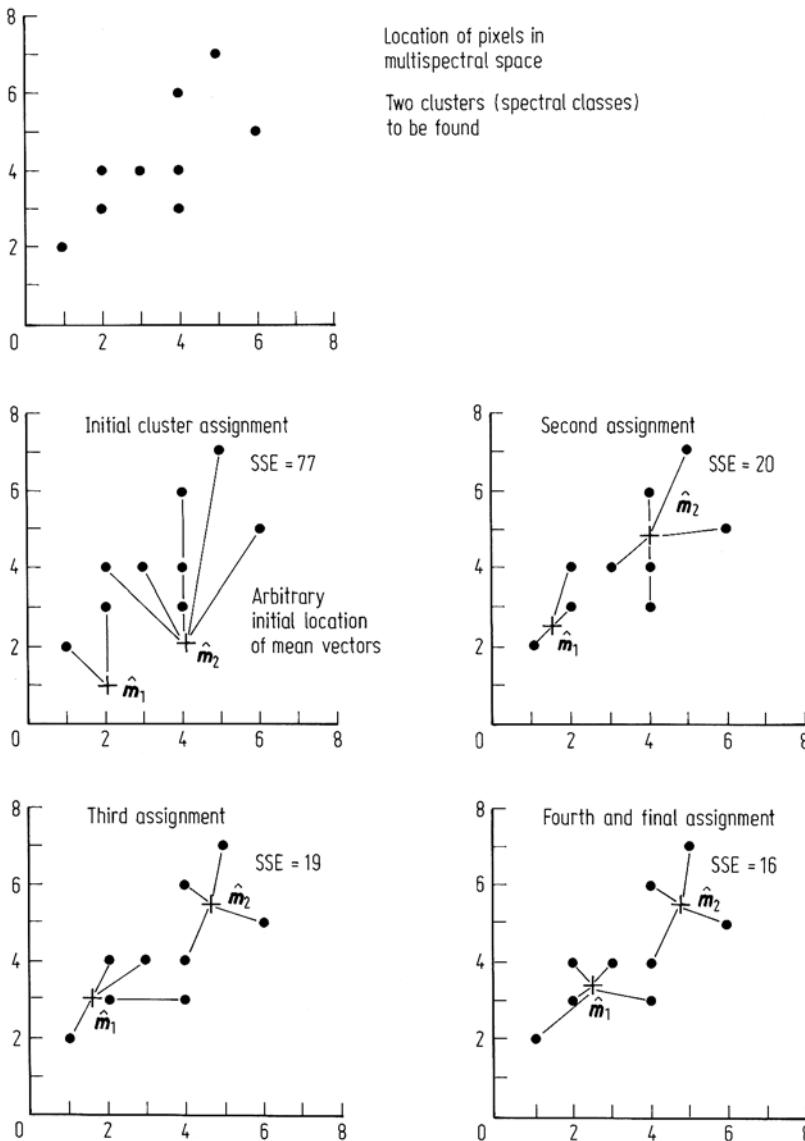
### 9.3.2

#### Mergings and Deletions

Once clustering is completed, or at any suitable intervening stage, the clusters can be examined to see whether

- (i) any clusters contain so few points as to be meaningless (e.g. that they would not give acceptable statistics estimates if used in training a maximum likelihood classifier), or
- (ii) some clusters are so close together that they represent an unnecessary or indeed an injudicious division of the data, and thus they should be merged.

In view of the material of Sect. 8.2.6 a guideline exists for (i), viz that a cluster would be of little value for training a maximum likelihood classifier if it did not contain about  $10N$  points where  $N$  is the number of spectral components. In Chap. 10, which deals with separability and divergence, means for deciding whether clusters should be merged can also be devised.



**Fig. 9.2.** An illustration of clustering by iterative optimization (or the isodata method). As noted, the method leads to a progressive reduction in SSE

**9.3.3****Splitting Elongated Clusters**

Another stage that can be inserted into the isodata algorithm is to separate elongated clusters into two new clusters. Usually this is done by prespecifying a standard deviation in each spectral band beyond which a cluster should be halved. Again this can be done after a set number of iterations, also specified by the user.

**9.3.4****Choice of Initial Cluster Centres**

Initialisation of the iterative optimization procedure requires specification of the number of clusters expected, along with their starting positions. In practice the actual or optimum number of clusters to choose will not be known. Therefore it is often chosen conservatively high, having in mind that resulting inseparable clusters can be consolidated after the process is completed, or at intervening iterations, if a merging operation is available.

The choice of the initial locations of the cluster centres is not critical although evidently it will have an influence on the time it takes to reach a final, acceptable clustering. Since no guidance is available in general, the following is a logical procedure (Phillips 1973). The initial cluster centres are chosen uniformly spaced along the multidimensional diagonal of the multispectral pixel space. This is a line from the origin to the point corresponding to the maximum brightness value in each spectral component (corresponding to 255 for 8 bit data, etc.). This choice can be refined if the user has some idea of the actual range of brightness values in each spectral component, say by having previously computed histograms. In that case the cluster centres would be initialised along a diagonal through the actual multidimensional extremities of the data.

Choice of the initial locations of clusters in the manner described is a reasonable and effective one since they are then well spread over the multispectral space in a region in which many spectral classes occur, especially for correlated data such as that corresponding to soils, rocks, concretes, etc.

**9.3.5****Clustering Cost**

Obviously the major limitation of the isodata technique is the need to prespecify the number of cluster centres. If this specification is too high then *a posteriori* merging can be used; however this is an expensive strategy. On the other hand, if too few are chosen initially then some multimodal spectral classes will result which, in turn, will prejudice ultimate classification accuracy.

Irrespective of whether too many or too few clusters are used, the isodata approach is computationally expensive since, at each iteration, every pixel must be checked against all cluster centres. Thus for  $C$  clusters and  $P$  pixels,  $PC$  distances have to be computed at each iteration and the smallest found. For  $N$  band data, each Euclidean

distance calculation will require  $N$  multiplications and  $N$  additions, ignoring the square root operation in (9.1) since that need not be carried out. Thus for 20 classes and 10,000 pixels, 100 iterations isodata clustering requires 20 million multiplications per band of data.

## 9.4

### Unsupervised Classification and Cluster Maps

At the completion of clustering, pixels within a given group are usually given a symbol to indicate that they belong to the same cluster or spectral class. Using these symbols a cluster map can be produced; this is a map corresponding to the image which has been clustered, but in which the pixels are represented by their symbol rather than by the original multispectral data. Sometimes only part of an image is used to form the cluster centres, but all pixels can be allocated to one of the clusters through, say, an minimum distance assignment.

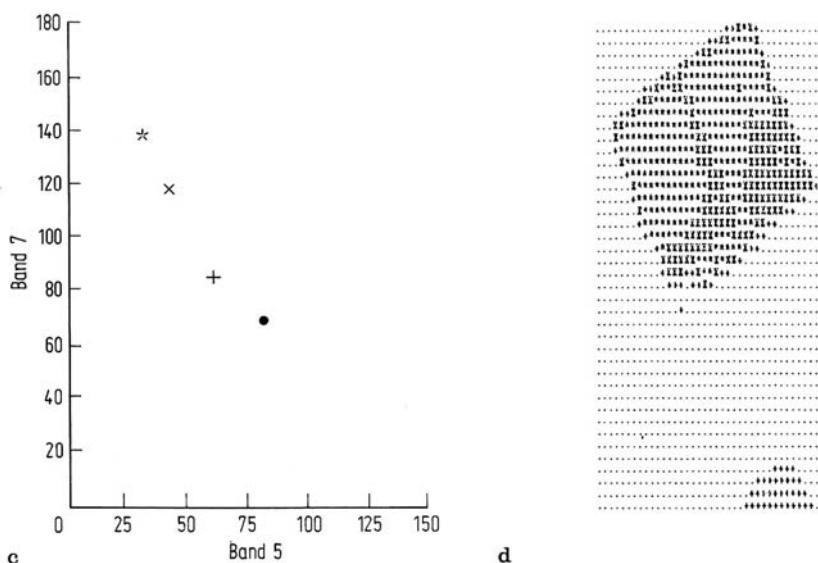
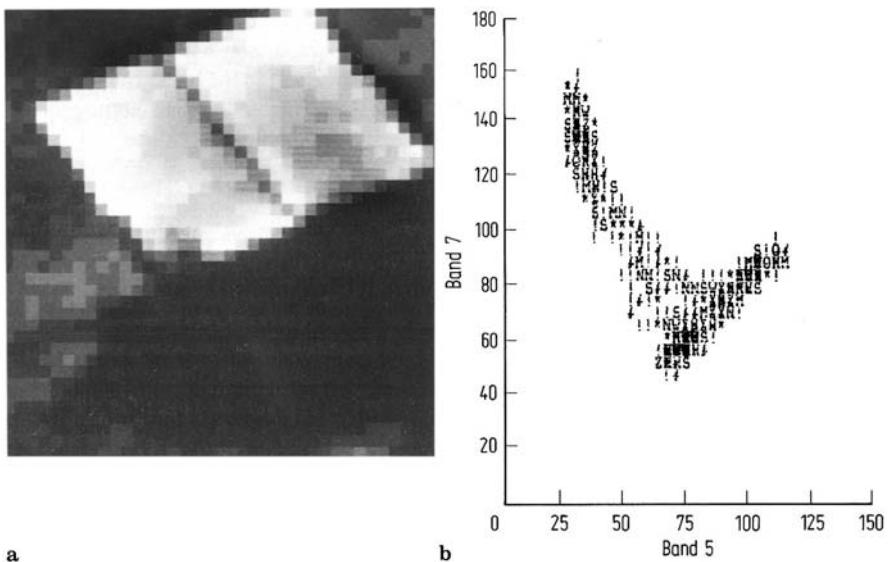
The availability of a cluster map allows a classification to be made. If some pixels with a given label can be identified with a particular ground cover type (by means of maps, site visits or other forms of reference data) then all pixels with the same label can be associated with that class. This method of image classification, depending as it does on *a posteriori* recognition of the classes, is called unsupervised classification since the analyst plays no part until the computational aspects are complete. Often unsupervised classification is used as a stand-alone technique, particularly when reliable training data for supervised classification cannot be obtained or is too expensive to acquire. However, it is also of value, as noted earlier, to determine the spectral classes that should be considered in a subsequent supervised approach. This is pursued in detail in Chap. 11.

## 9.5

### A Clustering Example

To illustrate the nature of the results produced by the iterative optimization algorithm a simple example with Landsat multispectral scanner data is presented. Figure 9.3a shows a small image segment (band 7 only for illustration) which consists of regions of crops and background soils. Figure 9.3b shows a scatter diagram for the image. In this, band 7 versus band 5 brightnesses of the pixels have been plotted. This is a subspace of the full four dimensional multispectral space of the image and gives an illustration of how the data points are distributed.

The data was clustered using the iterative optimization procedure (Kelly, 1983). Only five iterations were used and the algorithm was asked to determine five clusters. Merging and splitting options were employed at the end of each iteration leading ultimately to the four clusters shown on the plot of cluster means in Fig. 9.3c and to the cluster map shown in Fig. 9.3d. Comparison with Fig. 9.3a shows that the



**Fig. 9.3.** **a** Image segment used in the clustering illustration; **b** band 7 versus band 5 scatter diagram for the image; **c** cluster centres on a band 7 versus band 5 diagram; **d** cluster map produced by the isodata algorithm.

**Table 9.1.** Cluster means and standard deviations for Fig. 9.3. generated by the iterative optimization procedure

Cluster	Symbol	Band	Mean	St. Dev.
1	•	4	74.4	9.6
		5	85.5	13.7
		6	89.9	14.2
		7	69.8	12.1
2	*	4	45.0	2.0
		5	32.4	2.2
		6	127.4	6.3
		7	136.8	5.9
3	+	4	60.0	3.2
		5	59.5	4.0
		6	94.4	6.9
		7	83.7	7.5
4	×	4	48.9	3.8
		5	39.1	6.5
		6	114.0	5.9
		7	116.3	8.4

vegetation classes have been segmented more finely than the background soils in this case. Nevertheless the cluster map displays acceptable spatial homogeneity. Numerical details of the clusters established are given in Table 9.1.

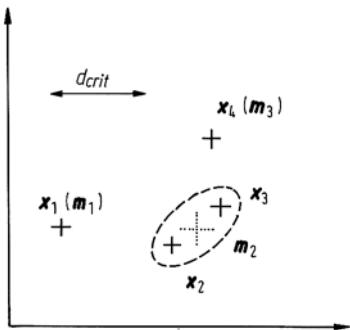
It is important to realise that the results generated in this example are not unique but depend upon the clustering parameters chosen. In practice the user may need to apply the algorithm several times with different parameter values to generate the desired segmentation.

## 9.6 A Single Pass Clustering Technique

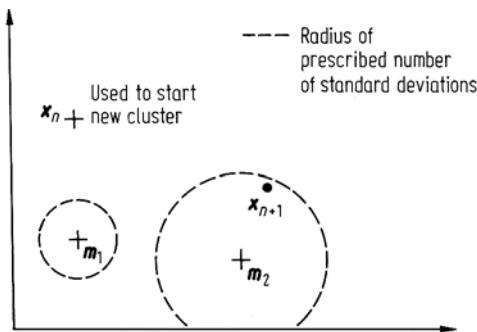
In order to reduce the cost of clustering image data, alternatives to iterative optimisation have been proposed and are widely implemented in software packages for remote sensing image analysis. Often what they gain in speed they may lose in accuracy; however if the user is aware of their characteristics they can usually be employed effectively. One fast clustering procedure which requires only a single pass through the data is described in the following subsection.

### 9.6.1 Single Pass Algorithm

Not all of the region to be clustered must be used in developing cluster centres but rather, for cost reduction, a randomly selected sample may be chosen and the samples



**Fig. 9.4.** Illustration of generation of cluster centres using the first row of samples



**Fig. 9.5.** Means by which pixels in the second and subsequent rows of samples are handled in the single pass clustering algorithm

arranged into a two dimensional array. The first row of samples is then used to obtain a starting set of cluster centres. This is initiated by adopting the first sample as the centre of the first cluster. If the second sample in the first row is further away from the first than a user specified *critical distance* then it is used to form another cluster centre. Otherwise the two samples are said to belong to the same cluster and their mean is computed as the new cluster centre. This procedure, which is illustrated in Fig. 9.4, is applied to all samples in the first row. Once this row has been exhausted the multidimensional standard deviations of the clusters are computed. Each sample in the second and subsequent rows is checked to see which cluster it is closest to. It is assigned to that cluster, and the cluster statistics recomputed, if it lies within a user-prescribed *number of standard deviations*. Otherwise it is used to form a new cluster centre (which is assigned a nominal standard deviation). This is depicted in Fig. 9.5. In this manner all of the samples are clustered and clusters with less than a prescribed number of pixels are deleted. Should a cluster map be required then the original segment of image data is scanned pixel by pixel and each pixel labelled according to the class it is closest to (on the basis usually of Euclidean distance). Should it be an outlying pixel in terms of the available cluster centres it is not labelled.

### 9.6.2

#### **Advantages and Limitations**

Apart from speed, a major advantage of this approach over the isodata procedure is its ability to create cluster centres as it proceeds. It is therefore not necessary for the user to specify beforehand the required number of clusters. However the method has two limitations. First, the user has to have a feel for the parameters required by the algorithm. In particular the user has to specify the critical distance parameter sensibly to enable the initial cluster centres to be established in a reasonable manner. Also the user has to know how many standard deviations should be used in assigning pixels in the second and subsequent lines of samples to existing clusters. Clearly, with experience, these parameters can be estimated reasonably.

The second limitation is that the method is dependent upon the first line of samples to initiate the clustering. Since it is only a one pass algorithm and has no feedback checking mechanism by way of iteration, its ultimate set of cluster centres can depend significantly on the character of the first line of samples.

### 9.6.3

#### **Strip Generation Parameter**

Adjacent pixels along a line frequently belong to the same cluster, as is to be expected, particularly for images of cultivated regions. A method therefore for enhancing the speed of clustering is to compare a pixel with its predecessor and assign it to the same cluster immediately if it is similar. The similarity check often used is quite straightforward, consisting of a check of the brightness difference in each spectral band. The difference allowable for two pixels to be considered part of the same cluster is called the strip generation parameter.

### 9.6.4

#### **Variations on the Single Pass Algorithm**

The technique outlined in the preceding section has a number of variations. For example, the initial cluster centres can be specified by the user or alternatively can be created from the data using a critical distance parameter as illustrated in Fig. 9.4. Moreover rather than use a multiplier of standard deviation for assigning pixels from the second and subsequent rows of samples, some algorithms proceed exactly as for the first row, with standard deviation information not used at all. Some algorithms use the  $L_1$  metric of (9.2), rather than Euclidean distance, and some check inter-cluster distances and merge if this is indicated; periodically small clusters can also be eliminated.

The package known as MultiSpec, also uses just critical distance parameters over the full range, although the user can specify a different critical distance for the second and later rows of samples (Landgrebe and Biehl, 2004).

### 9.6.5

#### An Example

As an illustration, the single pass procedure has been applied to the data of Fig. 9.3a. An initial critical distance of 15.0 was used, along with a standard deviation multiplier of 20.0 and a strip generation parameter of 1.0. The results produced are shown in Table 9.2 and Fig. 9.6. Two points are to be noted. First, different clusters have been found compared with those of the iterative optimization algorithm in Sect. 9.5. In this case there are two soil and two vegetation classes. Secondly, the essential spatial character of the classes has been produced with this algorithm even though the cluster centres generated are also at different locations in the multispectral space. Again, the procedure may need to be used interactively in practice to achieve a desired segmentation.

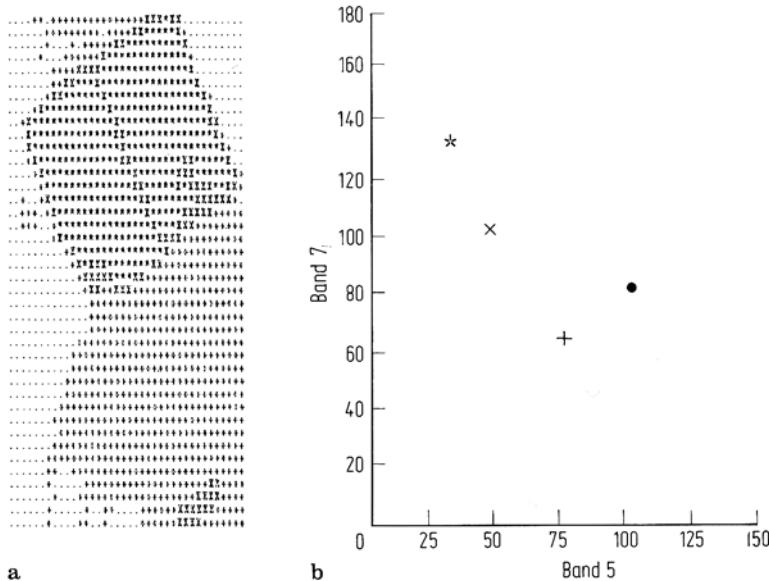
**Table 9.2.** Cluster means and standard deviations for Fig. 9.6. generated by the single pass algorithm

Cluster	Symbol	Band	Mean	St. Dev.
1	•	4	86.0	5.1
		5	102.6	6.1
		6	107.0	5.9
		7	84.2	4.6
2	+	4	68.6	5.6
		5	76.7	8.2
		6	82.6	9.1
		7	64.1	8.8
3	*	4	45.8	2.4
		5	33.7	3.1
		6	123.8	7.5
		7	131.3	9.3
4	×	4	53.4	4.8
		5	48.2	7.6
		6	105.5	6.2
		7	102.5	9.4

### 9.7

#### Agglomerative Hierarchical Clustering

Another clustering technique that does not require the user to specify the number of classes beforehand is hierarchical clustering. In fact this method produces an output that allows the user to decide the set of natural groupings into which the data falls. The procedure commences by assuming all pixels are individual clusters, it then systematically merges neighbouring clusters by checking distances between means. This is continued until all pixels appear in a single, larger cluster. An important aspect of the approach is that the history of mergings, or fusions as they are usually called



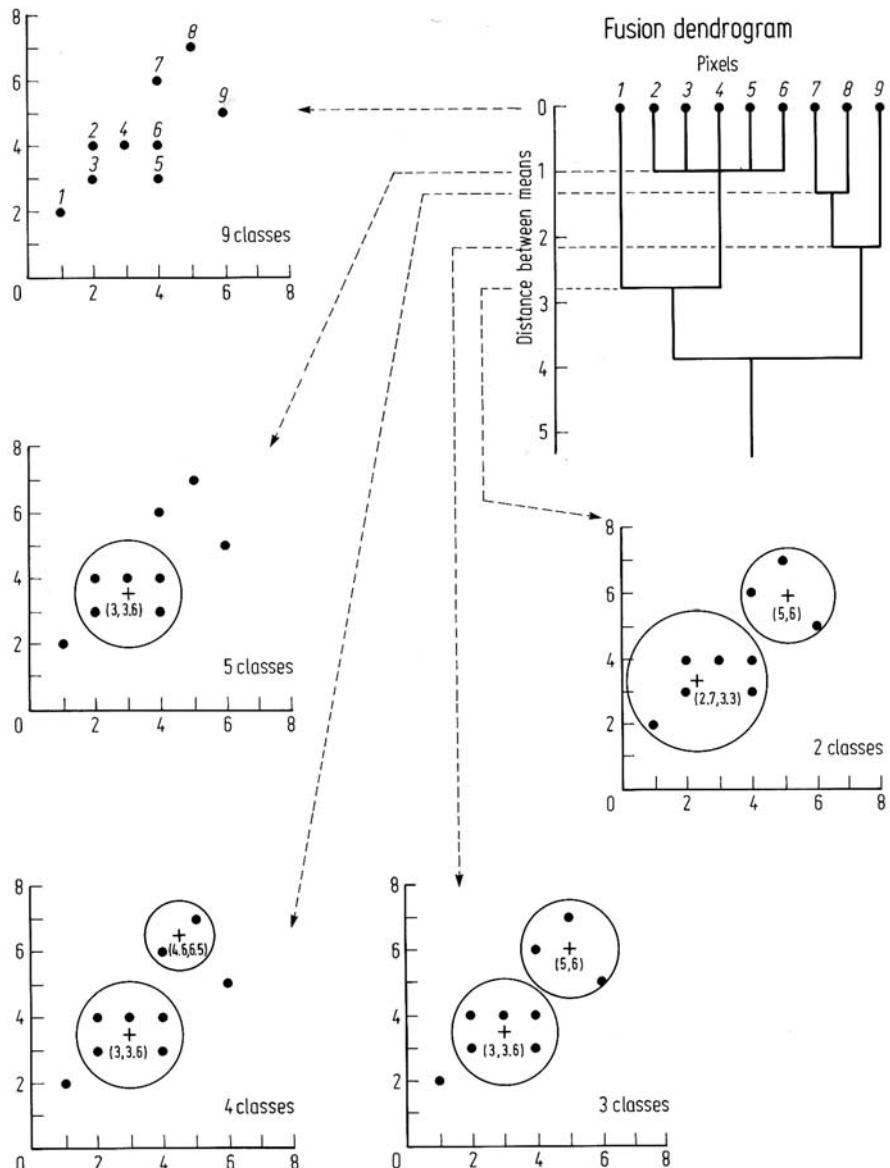
**Fig. 9.6.** **a** Cluster map and **b** cluster centres produced for the data of Fig. 9.3a, using the single pass clustering procedure

in this method, is displayed on a *dendrogram*. This is a diagram that shows at what distances between centres particular clusters are merged. An example of hierarchical clustering, along with its fusion dendrogram is shown in Fig. 9.7. This uses the same two dimensional data set as Fig. 9.2, but note that the ultimate cluster compositions are slightly different. This demonstrates again that different algorithms can and do produce different clusterings.

The fusion dendrogram of a particular hierarchical clustering exercise can be inspected in an endeavour to determine the intrinsic number of clusters or spectral classes in the data. Long vertical sections in the dendrogram between fusions indicate regions of “stability” which reflect natural data groupings. In Fig. 9.7 the longest region on the distance scale between fusions corresponds to two clusters in the data. One could conclude therefore that this data falls most naturally into two groups.

In the example presented, similarity between clusters was judged on the basis of Euclidean distance. Other similarity measures exist and are sometimes used, including divergence metrics as covered in Chap. 10.

The method given above is called *agglomerative* in view of its starting with a large number of clusters which it fuses progressively into a single cluster. *Divisive* hierarchical clustering procedures also exist in which the data is initialised as a single cluster which is progressively subdivided; these are more expensive computationally and are rarely used. Indeed hierarchical clustering generally does not find a lot of application in remote sensing image analysis since usually a large number of pixels is involved. Nevertheless it is a useful technique for small image data segments particularly since it can reveal data structure.



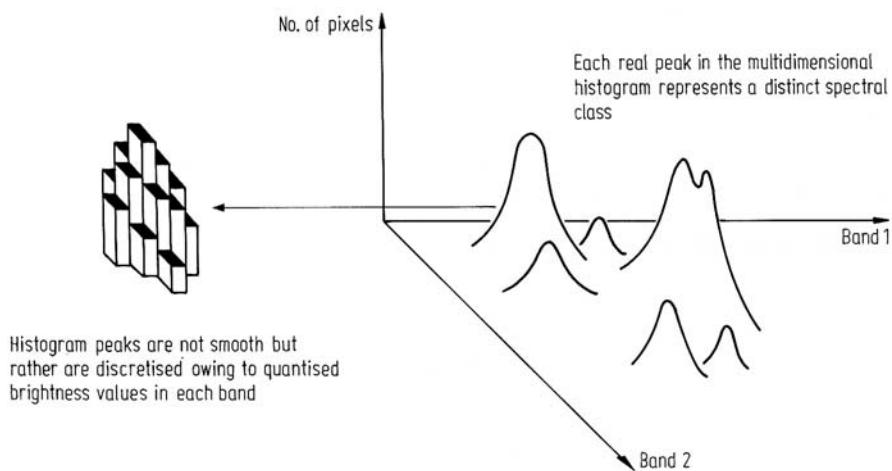
**Fig. 9.7.** An illustration of agglomerative hierarchical clustering, using Euclidean distance as a similarity measure

## 9.8

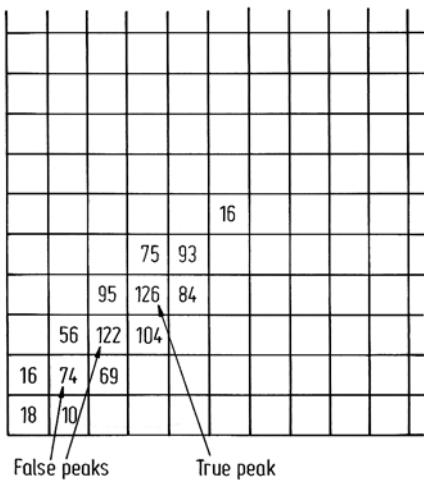
### Clustering by Histogram Peak Selection

A multidimensional histogram of a segment of image data may exhibit peaks at the locations of spectral classes or clusters. Consequently, a further clustering technique adopted with remote sensing data is to construct such a histogram and then search it to find the location of its peaks. Pixels are then associated with the nearest peak to produce the clusters. This method has been described by Letts (1978).

In using histogram peak selection as a clustering technique it is important to keep in mind that the data and the histogram are discrete in nature and not continuous, as shown in Fig. 9.8. To see the implications of this, consider the following calculation. A 100 pixel by 100 pixel image segment consists of 10,000 pixels. Suppose this corresponds to data with four spectral components each quantised into 256 levels of brightness. Then the corresponding four dimensional histogram will have  $(256)^4 = 4295$  million bins or locations into which counts (pixels) will be accumulated. If the bins were filled uniformly then a very sparse histogram would result. Indeed, on the average, there would be only one pixel per half a million bins. Each pixel therefore would appear as a local peak, which clearly would not be a true cluster. The bins of course would not be filled uniformly but nevertheless with bins only one brightness value wide in each spectral component, many artificial peaks will result from some isolated bins occupied by a single pixel and surrounded by empty bins. To circumvent this problem the histogram is accumulated with bins which are several brightness values wide in each dimension. In addition the dynamic range of the data in each dimension is ascertained beforehand from an inspection of the individual histograms in those dimensions. As an illustration, if the individual spectral component histograms for the four bands covered the ranges (35,95), (25,105), (20,80) and (5,65) and bin sizes of 10 brightness values were chosen for each dimension then the



**Fig. 9.8.** Illustration of a two dimensional histogram emphasising its discrete nature



**Fig. 9.9.** False indication of peaks in a two dimensional histogram, when the peak detection algorithm only searches parallel to the bins

total number of four dimensional bins is now  $6 \times 8 \times 6 \times 6 = 1728$ . With a  $100 \times 100$  pixel image segment therefore, there are, on the average, 6 pixels per bin which is probably acceptable (although low) to guarantee that peaks determined represent the location of real clusters in the data and not artifacts. Clearly resolution is sacrificed but this is necessary to yield an acceptable clustering by this approach.

The maximum detection algorithm used in this clustering procedure cannot be too sophisticated otherwise the method becomes too expensive to implement. Usually it consists of locating bins in which the count is higher than in the neighbouring bins along the same row and down the same column. For correlated data this can sometimes lead to false indications of peaks, as depicted in Fig. 9.9, in the vicinity of true peaks. This will be so particularly for smaller bin sizes. A better maximum detection procedure is to check diagonal neighbours as well but of course this doubles the search time.

Clearly this technique is only useful when the dimensionality of the data is low (just a few spectral bands). Because of the enormous number of bins that would be generated, and the extreme sparseness of the resulting histogram (see Problem 1.9), the method is not applicable to hyperspectral data sets.

## References for Chapter 9

Cluster analysis is a common tool in many applications that involve large amounts of data. Consequently source material on clustering algorithms will be found spread over many disciplines including numerical taxonomy, the social sciences and the physical sciences. However, because of the immense volumes of data to be clustered in remote sensing, the range of techniques that can be used is limited largely to those methods presented in this chapter and to their variations. Some more general treatments however that may be of value include Anderberg (1973), Hartigan (1975), Tryon and Bailey (1970) and Ryzin (1977).

- M.R. Anderberg, 1973: Cluster Analysis for Applications. N.Y. Academic.
- G.H. Ball and D.J. Hall, 1965: A Novel Method of Data Analysis and Pattern Classification. Stanford Research Institute, Menlo Park, California.
- G.R. Coleman and H.C. Andrews, 1979: Image Segmentation by Clustering. Proc. IEEE, 67, 773–785.
- R.D. Duda, P.E. Hart and D.G. Stork, 2001: Pattern Classification, 2e. N.Y., Wiley.
- J.A. Hartigan, 1975: Clustering Algorithms. N.Y., Wiley.
- D.J. Kelly, 1983: The Concept of a Spectral Class – A Comparison of Clustering Algorithms. M. Eng. Sc. Thesis. The University of New South Wales, Australia.
- D.A. Landgrebe and L. Biehl, 1995: An Introduction to MultiSpec. West Lafayette, Purdue Research Foundation (<http://dynamo.ecn.purdue.edu/~biehl/MultiSpec>
- P.A. Letts, 1978: Unsupervised Classification in The Aries Image Analysis System. Proc. 5th Canadian Symp. on Remote Sensing, 61–71.
- T.L. Phillips (Ed.), 1973: LARSYS Version 3 Users Manual. Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette.
- J. van Ryzin, 1977: Classification and Clustering. N.Y., Academic.
- R.C. Tryon and D.E. Bailey, 1970: Cluster Analysis, N.Y., McGraw-Hill.

## Problems

**9.1** Repeat the exercise of Fig. 9.2 but with

- (i) two initial cluster centres at (2,3) and (5,6),
- (ii) three initial cluster centres at (1,1), (3,3) and (5,5), and
- (iii) three initial cluster centres at (2,1), (4,2) and (15,15).

**9.2** From a knowledge of how a particular clustering algorithm works it is sometimes possible to infer the multidimensional spectral shapes of the clusters generated. For example, methods that depend entirely upon Euclidean distance as a similarity metric would tend to produce hyperspheroidal clusters. Comment on the cluster shapes you would expect to be generated by the migrating means technique based upon Euclidean distance and the single pass procedure, also based upon Euclidean distance.

**9.3** Suppose two different techniques have given two different clusterings of a particular set of data and you wish to assess which of the two segmentations is the better. One approach might be to evaluate the sum of square errors measure treated in Sect. 9.2. Another could be based upon covariance matrices. For example it is possible to define an “among clusters” covariance matrix that describes how the clusters themselves are scattered about the data space, and an average “within class” covariance matrix that describes the average shape and size of the clusters. Let these be called  $\Sigma_A$  and  $\Sigma_W$  respectively. How could they be used together to assess the quality of the two clustering results? (See Coleman and Andrews, 1979) Here you may wish to use measures of the “size” of a matrix, such as its trace or determinant (see Appendix D).

**9.4** Different clustering methods often produce quite different segmentations of the same set of data, as illustrated in the examples of Figs. 9.3 and 9.6. Yet the results generated for remote sensing applications are generally usable. Why do you think that is the case? (Hint: Is it related to the number of clusters generated?)

**9.5** The Mahalanobis distance of (8.13) can be used as the similarity metric for a clustering algorithm. Invent a possible clustering technique based upon (8.13) and comment on the nature of the clusters generated.

**9.6** Do you see value in having a two stage clustering process say in which a single pass procedure is used to generate initial clusters and then an iterative technique is used to refine them?

**9.7** Recompute the agglomerative hierarchical clustering example of Fig. 9.7 but use the  $L_1$  distance measure in (9.2) as a similarity metric.

**9.8** The histogram peak selection clustering technique of Sect. 9.8 has some shortcomings. One is related to the need to have large spectral bins in the histogram in order to have a sensible histogram produced when the data dimensionality is high. A consequence is that fine spectral resolution is sacrificed leading to loss of discrimination of spectral classes that are very close. Do you think good spectral discrimination could be regained by applying the technique several times over, on each subsequent occasion clustering just within one of the clusters found previously? Discuss the details of this approach.

**9.9** Consider the two dimensional data shown in Fig. 9.2, and suppose the three pixels at the upper right form one cluster and the remainder another cluster. Such an assignment might have been generated by some clustering algorithm other than iterative optimisation. Calculate the sum of squared error for this new assignment and compare with the value of 16 found in Fig. 9.2. Comment?

# 10

## Feature Reduction

### 10.1

#### Feature Reduction and Separability

Classification cost increases with the number of features used to describe pixel vectors in multispectral space – i.e. with the number of spectral bands associated with a pixel. For classifiers such as the parallelepiped and minimum distance procedures this is a linear increase with features; however for maximum likelihood classification, the procedure most often preferred, the cost increase with features is quadratic. Therefore it is sensible economically to ensure that no more features than necessary are utilised when performing a classification.

Section 8.2.6 draws attention to the number of training pixels needed to ensure that reliable estimates of class signatures can be obtained. In particular, the number of training pixels required increases with the number of bands or channels in the data. For high dimensionality data, such as that from imaging spectrometers, that requirement presents quite a challenge in practice, so keeping the number of features used in a classification to as few as possible is important if reliable results are to be expected from affordable numbers of training pixels.

Features which do not aid discrimination, by contributing little to the separability of spectral classes, should be discarded. Removal of least effective features is referred to as feature selection, this being one form of feature reduction. The other is to transform the pixel vector into a new set of coordinates in which the features that can be removed are made more evident. Both procedures are considered in some detail in this chapter.

Feature selection cannot be performed indiscriminantly. Methods must be devised that allow the relative worths of features to be assessed in a quantitative and rigorous way. A procedure commonly used is to determine the mathematical *separability* of classes; in particular, feature reduction is performed by checking how separable various spectral classes remain when reduced sets of features are used. Provided separability is not lowered unduly by the removal of features then those features can be considered of little value in aiding discrimination.

## 10.2

### Separability Measures for Multivariate Normal Spectral Class Models

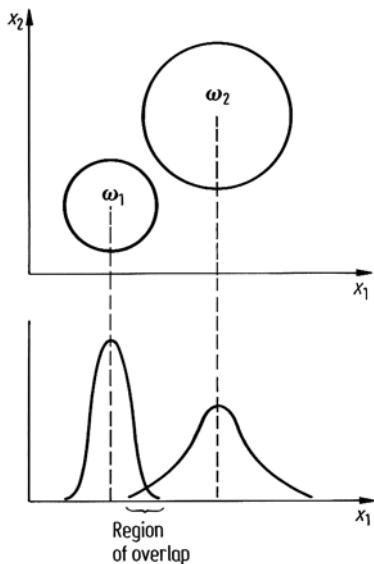
(Adapted in part from Swain and Davis, 1978)

#### 10.2.1

##### Distribution Overlaps

Consider a two dimensional multispectral space with two spectral classes as depicted in Fig. 10.1. Suppose we wish to see whether the classes could be separated using only one feature – either  $x_1$  or  $x_2$ . Of course it is not known which feature offers the best prospects *a priori*. This is what has to be determined by a measure of separability. Consider an assessment of  $x_1$ . The spectral classes in the  $x_1$  ‘subset’ or subspace are shown in the figure whereupon some overlap of the single dimensional distributions is indicated. If the distributions are well separated in the  $x_1$  dimension then clearly the overlap will be small and it would be unlikely that a classifier would make an error in discriminating between them on the basis of that feature alone. On the other hand for a large degree of overlap substantial classifier error would be expected. The usefulness of the  $x_1$  feature subset therefore can be assessed in terms of the overlap of the distributions in that domain, or more generally, in terms of the similarity of the distributions as a function of  $x_1$  alone.

Consider now an attempt to quantify the separation between a pair of probability distributions (as models of spectral classes) as an indication of the degree of overlap. Clearly distance between means is insufficient since overlap will also be influenced by the standard deviations of the distributions. Instead, a combination of both the



**Fig. 10.1.** Two dimensional multispectral space showing a hypothetical degree of separation possible in a single dimension subspace (in which class densities are shown)

distance between means and a measure of standard deviation is required. Moreover this must be a vector-based measure in order to be applicable to multidimensional subspaces. Several such measures are available; only those commonly encountered in connection with remote sensing data are treated in this chapter. Others may be found in books on statistics that treat similarities in probability distributions. These measures are all referred to as measures of separability which implies the ease with which patterns can be correctly associated with their classes using statistical pattern classification.

## 10.2.2 Divergence

### 10.2.2.1 A General Expression

Divergence is a measure of the separability of a pair of probability distributions that has its basis in their degree of overlap. It is defined in terms of the likelihood ratio

$$L_{ij}(\mathbf{x}) = p(\mathbf{x}|\omega_i)/p(\mathbf{x}|\omega_j)$$

where  $p(\mathbf{x}|\omega_i)$  and  $p(\mathbf{x}|\omega_j)$  are the values of the  $i$ th and  $j$ th spectral class probability distributions at the position  $\mathbf{x}$ . These are shown in an overlap region in Fig. 10.2 whereupon it is evident that  $L_{ij}(\mathbf{x})$  is a measure of ‘instantaneous’ overlap. Clearly for very separable spectral classes  $L_{ij}(\mathbf{x}) = 0$  or  $\infty$  for all  $\mathbf{x}$ .

It is of value to choose the logarithm of the likelihood ratio, viz

$$L'_{ij} = \ln p(\mathbf{x}|\omega_i) - \ln p(\mathbf{x}|\omega_j),$$

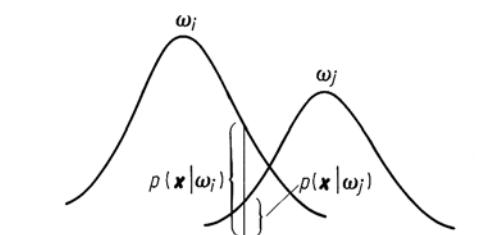
by means of which the *divergence* of the pair of class distribution is defined as

$$\mathcal{E}\{L'_{ij}(\mathbf{x})|\omega_i\} + \mathcal{E}\{L'_{ji}(\mathbf{x})|\omega_j\} \quad (10.1)$$

where  $\mathcal{E}\{\}$  is the expectation operator defined for continuous distributions as

$$\mathcal{E}\{L'_{ij}(\mathbf{x})|\omega_i\} = \int_x L'_{ij}(\mathbf{x}) p(\mathbf{x}|\omega_i) d\mathbf{x}.$$

This is the average or expected value of the likelihood ratio with respect to all patterns



**Fig. 10.2.** Definition of the probabilities used in the likelihood ratio

in the  $i$ th spectral class. Similarly for  $\mathcal{E}\{L'_{ji}(\mathbf{x})|\omega_j\}$ . From (10.1) it can be seen that

$$d_{ij} = \int_{\mathbf{x}} \{p(\mathbf{x}|\omega_i) - p(\mathbf{x}|\omega_j)\} \ln \frac{p(\mathbf{x}|\omega_i)}{p(\mathbf{x}|\omega_j)} d\mathbf{x}$$

from which a number of properties of divergence can be established. For example it is always positive and also  $d_{ji} = d_{ij}$ , as should be the case – i.e., it is symmetric. Moreover, if  $p(\mathbf{x}|\omega_i) = p(\mathbf{x}|\omega_j)$  for all  $\mathbf{x}$  then  $d_{ij} = d_{ji} = 0$  – in other words there is no divergence (or difference) between a distribution and itself.

For statistically independent features (i.e., spectral components)  $x_1, x_2, \dots, x_N$  then

$$p(\mathbf{x}|\omega_i) = \prod_{n=1}^N p(x_n|\omega_i)$$

which leads to

$$d_{ij}(\mathbf{x}) = \sum_{n=1}^N d_{ij}(x_n).$$

Since divergence is never negative it follows therefore that

$$d_{ij}(x_1, \dots, x_n, x_{n+1}) > d_{ij}(x_1, \dots, x_n).$$

In other words, divergence never decreases as the number of features is increased.

The material to this point has been general, applying to any multivariate spectral class model.

### 10.2.2.2

#### Divergence of a Pair of Normal Distributions

Since spectral classes in remote sensing image data are modelled by multidimensional normal distributions it is of particular interest to have available the specific form of (10.1) when  $p(\mathbf{x}|\omega_i)$  and  $p(\mathbf{x}|\omega_j)$  are normal distributions with means and covariances of  $\mathbf{m}_i$ ,  $\Sigma_i$  and  $\mathbf{m}_j$ ,  $\Sigma_j$  respectively.

By substitution of the full expressions for the normal distributions it can be shown that

$$\begin{aligned} d_{ij} &= \frac{1}{2} T_r \left\{ (\Sigma_i - \Sigma_j)(\Sigma_j^{-1} - \Sigma_i^{-1}) \right\} \\ &\quad + \frac{1}{2} T_r \left\{ (\Sigma_i^{-1} + \Sigma_j^{-1})(\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^t \right\} \\ &= \text{Term 1} + \text{Term 2}. \end{aligned} \tag{10.2}$$

where  $T_r\{\}$  is the trace of the subject matrix. Note that Term 1 involves only covariances whereas Term 2 is the square of a normalised (by covariance) distance between the means of the distributions.

Equation (10.2) gives the divergence between a *pair* of spectral classes that are normally distributed. Should there be more than two spectral classes, as is generally

the case, all pairwise divergences need to be checked to see whether a particular feature subset gives sufficiently separable data. An average indication of separability is then given by computing the *average divergence*

$$d_{ave} = \sum_{i=1}^M \sum_{j=i+1}^M p(\omega_i) p(\omega_j) d_{ij} \quad (10.3)$$

where  $M$  is the number of spectral classes and  $p(\omega_i)$ ,  $p(\omega_j)$  are the class prior probabilities.

### 10.2.2.3 Use of Divergence for Feature Selection

Consider the need to select the best three discriminating channels for Landsat multispectral scanner data, for an image in which only three spectral classes exist. The pairwise divergence between each pair of spectral classes would therefore be determined for all combinations of three out of four channels or bands. The feature subset chosen would be that which gives the highest overall indication of divergence – presumably this would be the highest average divergence. Table 10.1 illustrates the number of divergence calculations required for such an example.

In general, for  $M$  spectral classes,  $N$  total features, and a need to select the best  $n$  feature subset, the following set of pairwise divergence calculations are necessary, leaving aside the need finally to compute the average divergence for each subset.

First there are  ${}^N C_n$  possible combinations of  $n$  features from the total  $N$ , and for each combination there are  ${}^M C_2$  pairwise divergence measures to be computed. For a complete evaluation therefore

$${}^N C_n \cdot {}^M C_2$$

measures of pairwise divergence have to be calculated. To assess the best 4 of 7 Landsat Thematic Mapper bands for an image involving 10 spectral classes then

$${}^7 C_4 \cdot {}^{10} C_2 = 1575$$

divergence values have to be computed. Inspection of (10.2) shows each divergence calculation to be considerable. This, together with the large number required in a typical problem, makes the use of divergence to check separability and indeed separability analysis in general, an expensive process computationally.

**Table 10.1.** Divergence calculation table

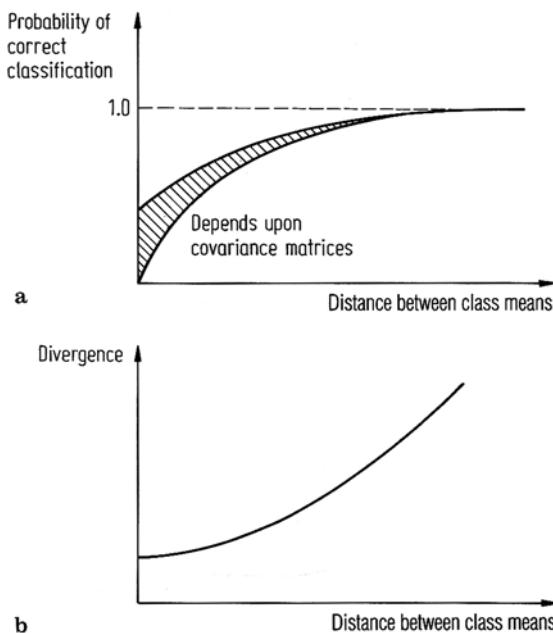
For channel subsets	$d_{12}$	$d_{13}$	$d_{23}$	$d_{ave}$
1, 2, 3	*	*	*	*
1, 2, 4	*	*	*	*
1, 3, 4	*	*	*	*
2, 3, 4	*	*	*	*

\* Entries to be calculated

### 10.2.2.4

#### A Problem with Divergence

As spectral classes become further removed from each other in multispectral space, the probability of being able to classify a pattern at a particular location moves asymptotically to 1.0 as depicted in Fig. 10.3a. If divergence is similarly plotted it will be seen from its definition that it increases quadratically with separation between spectral class means as depicted in Fig. 10.3b. This behaviour unfortunately is quite misleading if divergence is to be used as an indication of how successfully patterns in the corresponding spectral classes could be mutually discriminated or classified. It implies, for example, that at large separations, further small increases will lead to vastly better classification accuracy whereas in practice this is not the case as observed from the very slight increase in probability of correct classification implied by Fig. 10.3a. Moreover, outlying, easily separable classes will weight average divergence upwards in a misleading fashion to the extent that sub-optimal reduced feature subsets might be indicated as best, as illustrated in Swain and Davis (1978). This problem renders divergence, as it is presently defined, to be unsuitable and indeed unsatisfactory. The Jeffries-Matusita distance in the next section does not suffer this drawback.



**Fig. 10.3.** a Probability of correct classification as a function of spectral class separation; b divergence as a function of spectral class separation

### 10.2.3

#### The Jeffries-Matusita (JM) Distance

##### 10.2.3.1 Definition

The JM distance between a pair of probability distributions (spectral classes) is defined as

$$J_{ij} = \int_x \{\sqrt{p(x|\omega_i)} - \sqrt{p(x|\omega_j)}\}^2 dx \quad (10.4)$$

which is seen to be a measure of the average distance between the two class density functions (Wacker, 1971). For normally distributed classes this becomes

$$J_{ij} = 2(1 - e^{-B}) \quad (10.5)$$

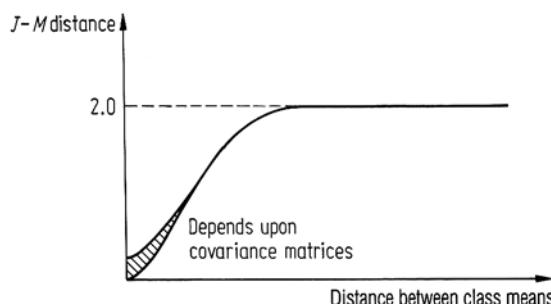
in which

$$\begin{aligned} B = & \frac{1}{8}(\mathbf{m}_i - \mathbf{m}_j)^t \left\{ \frac{\Sigma_i + \Sigma_j}{2} \right\}^{-1} (\mathbf{m}_i - \mathbf{m}_j) \\ & + \frac{1}{2} \ln \left\{ \frac{|(\Sigma_i + \Sigma_j)/2|}{|\Sigma_i|^{1/2} |\Sigma_j|^{1/2}} \right\} \end{aligned} \quad (10.6)$$

which is referred to as the Bhattacharyya distance (Kailath, 1967).

It is of interest to note that the first term in  $B$  is akin to the square of the normalised distance between the class means. The presence of the exponential factor in (10.5) gives an exponentially decreasing weight to increasing separations between spectral classes. If plotted as a function of distance between class means it shows a saturating behaviour not unlike that expected for the probability of correct classification, as seen in Fig. 10.4.

It is asymptotic to 2.0 so that a JM distance of 2.0 between spectral classes would imply classification of pixel data into those classes, (assuming they were the only two) with 100% accuracy. This saturating behaviour is highly desirable since it does not suffer the difficulty experienced with divergence.



**Fig. 10.4.** Jeffries-Matusita distance as a function of separation between spectral class means

As with divergence, an average pairwise JM distance can be defined according to

$$d_{ave} = \sum_{i=1}^M \sum_{j=i+1}^M p(\omega_i) p(\omega_j) J_{ij} \quad (10.7)$$

where  $M$  is the number of spectral classes and  $p(\omega_i), p(\omega_j)$  are the class prior probabilities.

### 10.2.3.2

#### Comparison of Divergence and JM Distance

JM distance performs better as a feature selection criterion for multivariate normal classes than divergence for the reasons given above; however it is computationally more complex and thus expensive to use as can be assessed from comparison of (10.2) and (10.6). Suppose a particular problem involves  $M$  spectral classes. Consider the cost then of computing all pairwise divergences and all pairwise JM distances. These costs can be assessed largely on the basis of having to compute matrix inverses and determinants, assuming reasonably that they involve similar computational demands using numerical procedures. In the case of divergence it is necessary to compute only  $M$  matrix inverses to allow all the pairwise divergences to be found. However for JM distance it is necessary to compute  ${}^M C_2 + M$  equivalent matrix inverses since the individual class covariances appear as pairs which have to be added and then inverted. It may be noted that  ${}^M C_2 + M = \frac{1}{2}M(M + 1)$  so that divergence is a factor of  $\frac{1}{2}(M + 1)$  more economical to use. When it is recalled how many feature subsets may need to be checked in a feature selection exercise this is clearly an important consideration. However the unbound nature of divergence as discussed in Sect. 10.2.2.4 throws doubt on its usefulness.

### 10.2.4

#### Transformed Divergence

##### 10.2.4.1

##### Definition

A useful modification of divergence becomes apparent by noting the algebraic similarity of divergence to the parameter  $B$  in JM distance, as defined in (10.6). Since both involve terms which are functions of the covariance alone, and terms which appear as normalised distances between class means, it should be possible to make use of a heuristic *transformed divergence* measure of the form (Swain and Davis 1978)

$$d_{ij}^T = 2(1 - e^{-d_{ij}/8}). \quad (10.8)$$

Because of its exponential character it will have a saturating behaviour with increasing class separation, as does JM distance, and yet it is computationally more

economical. This saturating measure is used in the software package called Multi-Spec; it has been demonstrated to be almost as effective as JM distance in feature selection, and considerably better than simple divergence or simple Bhattacharyya distance (Swain et al., 1971, Mausel et al., 1990).

#### 10.2.4.2

#### Relation Between Transformed Divergence and Probability of Correct Classification

It can be shown that the probability of making a classification error in placing a pattern into one of two (equal prior probability) classes with a pairwise divergence  $d_{ij}$  is bound by (Kailath, 1967)

$$p_E > \frac{1}{8} e^{-d_{ij}/2},$$

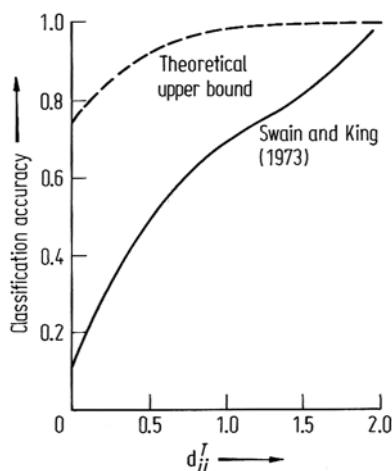
so that the probability of correct classification is bound by

$$p_C < 1 - \frac{1}{8} e^{-d_{ij}/2}.$$

Since  $d_{ij} = -8 \ln \left(1 - \frac{1}{2} d_{ij}^T\right)$  from (10.8),

$$\text{then } P_C < 1 - \frac{1}{8} \left(1 - \frac{1}{2} d_{ij}^T\right)^4. \quad (10.9)$$

This bound on classification accuracy is shown in Fig. 10.5 along with an empirical relationship between transformed divergence and probability of correct (pairwise) classification derived by Swain and King (1973). This figure has considerable value in establishing *a priori* the upper bound achievable on classification accuracy for an existing set of spectral classes.



**Fig. 10.5.** Probability of correct classification as a function of pairwise transformed divergence. The empirical measure, taken from Swain and King (1973), was determined using 2790 sets of multidimensional, normally distributed data, in two classes

### 10.2.4.3

#### Use of Transformed Divergence in Clustering

One of the last stages in a practical clustering algorithm is to evaluate the size and relative locations of the clusters produced, as noted in Chap. 9. If clusters are too close to each other they should be merged. The availability of the information in Fig. 10.5 allows merging to be effected based upon a pre-specified transformed divergence, since both cluster mean and covariance data is normally available. By establishing a desired accuracy level (in fact upper bound) for the subsequent classification and then determining the corresponding value of transformed divergence, clusters with separabilities less than this value must be merged.

## 10.3

### Separability Measures for Minimum Distance Classification

The separability measures of Sect. 10.2 relate to spectral classes modelled by multivariate normal distributions, in preparation for maximum likelihood classification. Should another classifier be used this procedure is unduly complex and largely without meaning. For example, if supervised classification is to be carried out using the minimum distance to class means technique there is no advantage in using distribution-based separability measures, since probability distribution class models are not employed. Instead it is better to use a simple measure consistent with the nature of the classification algorithm. For minimum distance calculation this would be a distance measure, computed according to the particular distance metric in use. Commonly this is Euclidean distance. Consequently, when a set of spectral classes has been determined, ready for the classification step, the complete set of pairwise Euclidean distances will provide an indication of class similarities. Unfortunately this cannot be related to an error probability (for misclassification) but finds application as an *indicator* of what pairs of classes could be merged, if so desired.

## 10.4

### Feature Reduction by Data Transformation

The emphasis of the preceding sections has been *feature selection* – i.e., an evaluation of the existing set of features for the pixel data in multispectral imagery with a view to selecting the most discriminating, and discarding the rest. It is also possible to effect feature reduction by transforming the data to a new set of axes in which separability is higher in a subset of the transformed features than in any subset of the original data. This allows transformed features to be discarded. A number of image transformations could be entertained for this; however the most commonly encountered in remote sensing are the principal components or Karhunen-Loëve transform and the transformation associated with so-called canonical analysis. These are treated in the following.

### 10.4.1

#### Feature Reduction Using the Principal Components Transformation

The principal components transformation (see Chap. 6) maps image data into a new, uncorrelated co-ordinated system or vector space. Moreover, in doing so, it produces a space in which the data has most variance along its first axis, the next largest variance along a second mutually orthogonal axis, and so on. The later principal components would be expected, in general, to show little variance. These could be considered therefore to contribute little to separability and could be ignored, thereby reducing the essential dimensionality of the classification space and thus improving classification speed. This is only of value however if the spectral class structure of the data is distributed substantially along the first few axes. Should this not be the case it is possible that feature reduction of the transformed data may be no more likely than with the original data. In such a case the technique of canonical analysis may be a better approach.

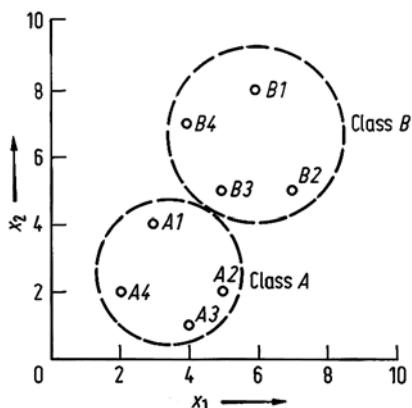
As an illustration of a situation of data in which principal components transformation does allow feature reduction, consider the two class two dimensional data illustrated in Fig. 10.6. Assume that the classes are not separable in either of the original data variables alone but rather both dimensions are required for separability. However, inspection indicates that the first component of a principal components transform will yield class separability. This is now demonstrated mathematically by presenting the results of hand calculations on the data.

Notwithstanding the class structure of the data the principal components transformation makes use of a global mean and global covariance. Using (6.1) and (6.2) it is shown readily that

$$\mathbf{m} = \begin{bmatrix} 4.5 \\ 4.25 \end{bmatrix}$$

and

$$\Sigma = \begin{bmatrix} 2.57 & 1.86 \\ 1.86 & 6.21 \end{bmatrix}$$



**Fig. 10.6.** Two dimensional, two class data in which feature reduction using principal components analysis is possible

The eigenvalues of the covariance matrix are  $\lambda_1 = 6.99$  and  $\lambda_2 = 1.79$  so that the first principal component will contain 79.6% of the variance. The normalised eigenvectors corresponding to these eigenvalues are

$$\mathbf{g}_1 = \begin{bmatrix} 0.387 \\ 0.922 \end{bmatrix} \quad \text{and} \quad \mathbf{g}_2 = \begin{bmatrix} -0.922 \\ 0.387 \end{bmatrix}$$

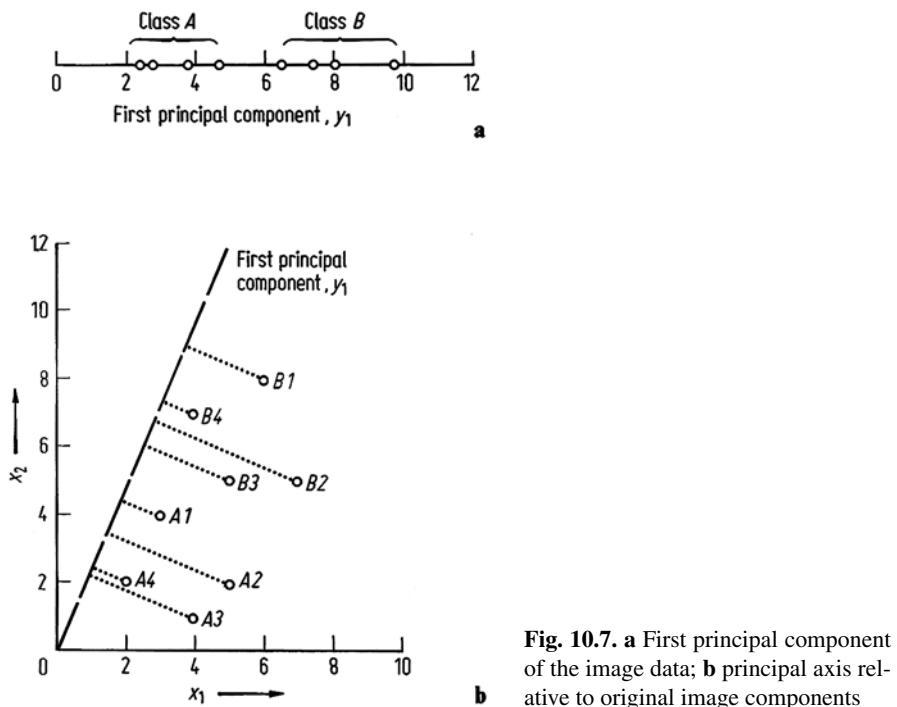
so that the principal components transformation matrix is

$$G = \begin{bmatrix} 0.387 & 0.922 \\ -0.922 & 0.387 \end{bmatrix} = D^t \text{ in (6.4).}$$

Using this matrix, the first principal component of each pixel vector can be computed according to

$$y_1 = 0.387x_1 + 0.922x_2.$$

These are shown plotted in Fig. 10.7a in which it is seen that the first principal component is sufficient for separation. Figure 10.7b shows the principal axes relative to the original image components.

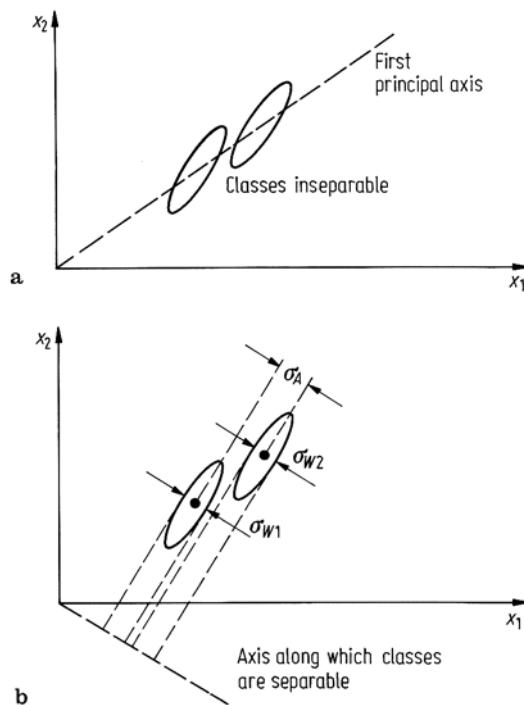


**Fig. 10.7. a** First principal component of the image data; **b** principal axis relative to original image components

### 10.4.2

#### Canonical Analysis as a Feature Selection Procedure

The principal components transformation is based upon the global covariance matrix of the full set of image data and thus is not sensitive explicitly to class structure in the data. The reason it often works well in remote sensing as a feature reduction tool is a result of the fact that classes are frequently distributed in the direction of maximum data scatter. This is particularly so for soils and spectrally similar cover types. Should good separation not be afforded by the principal components transformation derived from the global covariance matrix then a subset of image data could be selected that embodies the cover types of interest and this subset used to compute a covariance matrix. The resulting transformation will have its first principal axes oriented so that the cover types of interest are well discriminated. Another, more rigorous, method for generating a transformed set of feature axes, in which class separation is optimised, is based upon the procedure called canonical analysis. To illustrate this approach consider the contrived two dimensional, two class data shown in Fig. 10.8. By inspection, the classes can be seen not to be separable in either of the original feature axes on their own. Nor will they be separable in only one of the two principal component axes because of the nature of the global data scatter compared with the scatter of data within the individual classes.



**Fig. 10.8.** **a** Hypothetical two dimensional, two class data illustrating lack of separability in either original band or in either principal component; **b** axis along which classes can be separated

Inspection shows however that the data of Fig. 10.8a can be separated by a single feature if an axis rotation (i.e. an image transformation) such as that shown in Fig. 10.8b is adopted. A little thought reveals that the primary axis in this new transformation should be so-oriented that the classes have the largest possible separation between their means when projected onto that axis, while at the same time they should appear as small as possible in their individual spreads. If we characterise the former by a measure  $\sigma_A$  as illustrated in the diagram (which can be referred to as the standard deviation among the classes – it is as if the classes themselves were data points at their mean positions) and the spread of data within classes as seen on the new axis as  $\sigma_{w1}, \sigma_{w2}$  as illustrated (these are the standard deviations of the classes) then our interest is in finding a new axis for which

$$\frac{\sigma_A^2}{\sigma_w^2} = \frac{\text{among categories variance}}{\text{within categories variance}} \quad (10.10)$$

is as large as possible. Here  $\sigma_w^2$  is the average of  $\sigma_{w1}^2$  and  $\sigma_{w2}^2$  for the example of Fig. 10.8.

#### 10.4.2.1

#### Within Class and Among Class Covariance Matrices

To handle data with any number of dimensions it is necessary to define average data scatter within the classes, and the scatter of the classes themselves around the multispectral space, by covariance matrices.

The average within class covariance matrix is defined as

$$\Sigma_w = \frac{1}{M} \sum_{i=1}^M \Sigma_i \quad (10.11a)$$

where  $\Sigma_i$  is the covariance matrix of the data in class  $i$  and where  $M$  is the total number of classes. The boldface sigma is printed for the summation to distinguish it from the symbol for covariance. Equation (10.11a) applies only if the classes have equal populations. A better expression is

$$\Sigma_w = \left\{ \sum_{i=1}^M (n_i - 1) \Sigma_i \right\} / S_n \quad (10.11b)$$

where  $n_i$  is the population of the  $i$ th class and  $S_n = \sum_{i=1}^M n_i$ .

The among class covariance matrix is given by

$$\Sigma_A = \mathcal{E}\{(\mathbf{m}_i - \mathbf{m}_0)(\mathbf{m}_i - \mathbf{m}_0)^t\} \quad (10.12)$$

where  $\mathbf{m}_i$  is the mean of the  $i$ th class,  $\mathcal{E}$  is the expectation operator and  $\mathbf{m}_0$  is the global mean, given by

$$\mathbf{m}_0 = \frac{1}{M} \sum_{i=1}^M \mathbf{m}_i \quad (10.13a)$$

where the classes have equal populations, or

$$\mathbf{m}_0 = \sum_{i=1}^M n_i \mathbf{m}_i / S_n \quad (10.13b)$$

in general.

### 10.4.2.2 A Separability Measure

Let  $\mathbf{y} = D^t \mathbf{x}$  be the required transformation that generates the new axes  $\mathbf{y}$  in which the classes have optimal separation. The transposed form of the transformation matrix is chosen here to simplify the following expressions. By the same procedure that was used for the principal components transformation in Sect. 6.1.2 it is possible to show that the within class and among class covariance matrices in the new co-ordinate system are

$$\Sigma_{w,y} = D^t \Sigma_{w,x} D \quad (10.14a)$$

$$\Sigma_{A,y} = D^t \Sigma_{A,x} D \quad (10.14b)$$

where the subscripts  $x$  and  $y$  have been used to identify the matrices with their respective co-ordinates. It is significant to realise here, unlike with the case of principal components analysis, that the two new covariance matrices are not necessarily diagonal. However, as with principal components the row vectors of  $D^t$  define the axis directions in  $y$ -space. Let  $\mathbf{d}^t$  be one particular vector (say the one that defines the first so-called canonical axis, along which the classes will be optimally separated), then the corresponding within class and among class variances will be

$$\sigma_w^2 = \mathbf{d}^t \Sigma_{w,x} \mathbf{d}$$

$$\sigma_A^2 = \mathbf{d}^t \Sigma_{A,x} \mathbf{d} .$$

What we wish to do is to find the  $\mathbf{d}$ , (and in fact ultimately the full transformation matrix  $D^t$ ) for which

$$\lambda = \sigma_A^2 / \sigma_w^2 = \mathbf{d}^t \Sigma_{A,x} \mathbf{d} / \mathbf{d}^t \Sigma_{w,x} \mathbf{d} \quad (10.15)$$

is maximised. In the following the axis subscripts on the covariance matrices have been dropped for convenience.

### 10.4.2.3 The Generalised Eigenvalue Equation

The ratio of variances  $\lambda$  in (10.15) is maximised by the selection of  $\mathbf{d}$  if

$$\frac{\partial \lambda}{\partial \mathbf{d}} = 0.$$

Noting the identity that  $\frac{\partial}{\partial \mathbf{x}} \{ \mathbf{x}^t \mathbf{A} \mathbf{x} \} = 2 \mathbf{A} \mathbf{x}$  then

$$\begin{aligned}\frac{\partial \lambda}{\partial \mathbf{d}} &= \frac{\partial}{\partial \mathbf{d}} \{ (\mathbf{d}^t \Sigma_A \mathbf{d}) (\mathbf{d}^t \Sigma_w \mathbf{d})^{-1} \} \\ &= 2 \Sigma_A \mathbf{d} (\mathbf{d}^t \Sigma_w \mathbf{d})^{-1} - 2 \Sigma_w \mathbf{d} (\mathbf{d}^t \Sigma_A \mathbf{d}) (\mathbf{d}^t \Sigma_w \mathbf{d})^{-2} \\ &= 0.\end{aligned}$$

This reduces to

$$\Sigma_A \mathbf{d} - \Sigma_w \mathbf{d} (\mathbf{d}^t \Sigma_A \mathbf{d}) (\mathbf{d}^t \Sigma_w \mathbf{d})^{-1} = 0.$$

Which can be written as

$$(\Sigma_A - \lambda \Sigma_w) \mathbf{d} = 0 \quad (10.16)$$

Equation (10.16) is called a *generalised eigenvalue equation* and has to be solved now for the unknowns  $\lambda$  and  $\mathbf{d}$ . The first canonical axis will be in the direction of  $\mathbf{d}$  and  $\lambda$  will give the associated ratio of among class to within class variance along that axis.

In general (10.16) can be written

$$(\Sigma_A - \Lambda \Sigma_w) \mathbf{D} = 0 \quad (10.17)$$

where  $\Lambda$  is a diagonal matrix of the full set of  $\lambda$ 's and  $\mathbf{D}$  is the matrix of vectors  $\mathbf{d}$ .

The development to this stage is usually referred to as discriminant analysis. One additional step is included in the case of canonical analysis.

As with the equivalent step in the principal components transformation, solution of (10.16) amounts to finding the set of eigenvalues  $\lambda$  and the corresponding eigenvectors,  $\mathbf{d}$ . While unique values for  $\lambda$  can be determined the components of  $\mathbf{d}$  can only be found relative to each other. In the case of principal components we introduced the additional requirement that the vectors have unit magnitude, thereby allowing the vectors to be determined uniquely. For canonical analysis, the additional constraint used is

$$\mathbf{D}^t \Sigma_w \mathbf{D} = I. \quad (10.18)$$

This says that the within class covariance matrix after transformation must be the identity matrix (i.e. a unit diagonal matrix). In other words, after transformation, the classes should appear spherical.

For  $M$  classes and  $N$  bands of multispectral data, if  $N > M - 1$  there will only be  $M - 1$  non-zero roots of (10.17) and thus  $M - 1$  canonical axes (Seal, 1964). For this example, in which  $N = 2$ ,  $M = 2$ , one of the eigenvalues of (10.16) will be zero and thus the corresponding eigenvector will not exist. This implies that the dimensionality of the transformed space will be less than that of the original data. Thus canonical analysis provides separability with reduced dimensionality. In general, in the first canonical axis, corresponding to the largest  $\lambda$ , the classes will have maximum separation. The second axis, corresponding to the next largest  $\lambda$ , will provide the next best degree of separation, and so on. Campbell and Atchley (1981) review canonical analysis with a particular emphasis on a geometrical interpretation.

### 10.4.2.4 An Example

Consider the two dimensional, two category data shown in Fig. 10.9. Both of the original features  $x_1$  and  $x_2$  are required to discriminate between the categories. We will now perform a canonical analysis transformation on the data to show that the categories can be discriminated in the first canonical axis.

The individual covariance matrices of the classes are

$$\Sigma_A = \begin{bmatrix} 2.25 & 2.59 \\ 2.59 & 4.25 \end{bmatrix} \quad \Sigma_B = \begin{bmatrix} 4.25 & 3.00 \\ 3.00 & 6.67 \end{bmatrix}$$

so that the within class covariance is

$$\Sigma_w = \frac{1}{2}\{\Sigma_A + \Sigma_B\} = \begin{bmatrix} 3.25 & 2.80 \\ 2.80 & 5.46 \end{bmatrix}.$$

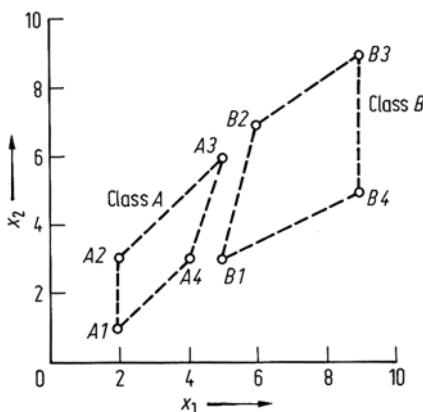
The among class covariance matrix is

$$\Sigma_A = \begin{bmatrix} 8.00 & 5.50 \\ 5.50 & 3.78 \end{bmatrix}.$$

The canonical transformation matrix  $D^t$  is given by a solution to (10.17) where  $D$  is a matrix of column vectors. These vectors are the axes in the transformed space, along the first of which the ratio of among categories variance to within categories variance is greatest. Along this axis there is most chance of separating the classes.  $\Lambda$  is a diagonal matrix of scalar constants that are the eigenvalues of (10.17); numerically these are the ratios of variances along each of the canonical axes.

Each  $\lambda$  and the accompanying  $d$  can be found readily by considering the individual component equation (10.16) rather than the more general form in (10.17). For (10.16) to have a non-trivial solution it is necessary that the determinant

$$|\Sigma_A - \lambda \Sigma_w| = 0.$$



**Fig. 10.9.** Two classes of two dimensional data, each containing 4 data points

Using the values for  $\Sigma_A$  and  $\Sigma_w$  above this is

$$\begin{vmatrix} 8.00 - 3.25\lambda & 5.50 - 2.80\lambda \\ 5.50 - 2.80\lambda & 3.78 - 5.46\lambda \end{vmatrix} = 0$$

which gives  $\lambda = 2.54$  or 0. Thus there is only one canonical axis defined by the vector  $\mathbf{d}$  corresponding to  $\lambda = 2.54$ . This is given as the solution to

$$[\Sigma_A - 2.54\Sigma_w]\mathbf{d} = 0$$

i.e.

$$\begin{bmatrix} -0.26 & -1.61 \\ -1.61 & -10.09 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 0$$

whereupon  $d_1 = -6.32d_2$ .

At this stage we use (10.18), which for one vector  $\mathbf{d}$  in  $D$  is

$$[d_1 \ d_2] \begin{bmatrix} 3.25 & 2.80 \\ 2.80 & 5.46 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} = 1$$

i.e.  $3.25 d_1^2 + 5.60 d_1 d_2 + 5.46 d_2^2 = 1$ .

Using  $d_1 = -6.32 d_2$  gives

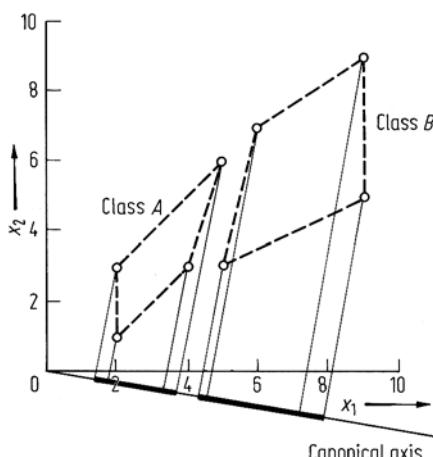
$$d_1 = 0.632, \ d_2 = -0.100$$

so that

$$\mathbf{d} = \begin{bmatrix} 0.632 \\ -0.100 \end{bmatrix}$$

This vector is shown plotted in Fig. 10.10 wherein the projections of the patterns onto the axis defined by that vector show the classes to be separable. The brightness values of the pixels in that first axis are given by

$$\begin{aligned} y &= \mathbf{d}^T \mathbf{x} \\ &= [0.632 \ -0.100] \mathbf{x}. \end{aligned}$$



**Fig. 10.10.** The first canonical axis for the two class data of Fig. 10.9 showing class discrimination

### 10.4.3

#### Discriminant Analysis Feature Extraction (DAFE)

A variation on the canonical analysis development of the previous section is to form the Fisher criterion

$$J = \text{Tr} \left\{ \Sigma_{w,y}^{-1} \Sigma_{A,y} \right\} \quad (10.19)$$

rather than the measure of (10.15). We want to find an axis transformation that minimises  $J$ . Let the transformation be  $\mathbf{y} = D^t \mathbf{x}$ . Then (10.19) can be written

$$J = \text{Tr} \left\{ (D^t \Sigma_{w,x} D)^{-1} (D^t \Sigma_{A,x} D) \right\} .$$

It is shown by Fukunaga (1990) that differentiating this last expression to find the transformation matrix  $D^t$  that minimises  $J$  leads to

$$\Sigma_{w,x}^{-1} \Sigma_{A,x} D = D \Sigma_{w,y}^{-1} \Sigma_{A,y}. \quad (10.20)$$

Consider the transformation  $\mathbf{z} = B^t \mathbf{y}$  that diagonalises the transformed among class covariance  $\Sigma_{A,y}$ :

$$B^t \Sigma_{A,y} B = M .$$

in which  $M$  is diagonal. Thus

$$\Sigma_{A,y} = B^{t^{-1}} M B^{-1}$$

so that (10.20) becomes

$$\Sigma_{w,x}^{-1} \Sigma_{A,x} D = D \Sigma_{w,y}^{-1} B^{t^{-1}} M B^{-1} \quad (10.21)$$

As with canonical analysis we now introduce the additional criterion that the within class covariance matrix be unity after the transformation to  $z$  space, so that the classes then appear hyperspherical. This requires

$$B^t \Sigma_{w,y} B = I$$

$$\text{or } B^{-1} \Sigma_{w,y}^{-1} B^{t^{-1}} = I$$

so that  $\Sigma_{w,y}^{-1} B^{t^{-1}} = B$  which, when substituted into (10.21),

$$\text{gives } \Sigma_{w,x}^{-1} \Sigma_{A,x} D = DBMB^{-1}$$

$$\text{or } \Sigma_{w,x}^{-1} \Sigma_{A,x} DB = DBM$$

which we recognise as an eigenfunction equation in which  $M$  is a diagonal matrix of the eigenvalues of  $\Sigma_{w,x}^{-1} \Sigma_{A,x}$  and  $(DB)$  is the matrix of eigenvectors of  $\Sigma_{w,x}^{-1} \Sigma_{A,x}$ . Eigenanalysis of  $\Sigma_{w,x}^{-1} \Sigma_{A,x}$  can be carried out by analysis of  $\Sigma_{w,x}^{-1}$  and  $\Sigma_{A,x}$  separately (Fukunaga, 1990).

The axis along which the data has maximum separability is that corresponding to the largest eigenvalue of  $\Sigma_{w,x}^{-1} \Sigma_{A,x}$ , and so on. Deriving the transformed axes based on minimisation of (10.19) is referred to as discriminant analysis feature extraction (DAFE).

#### 10.4.4

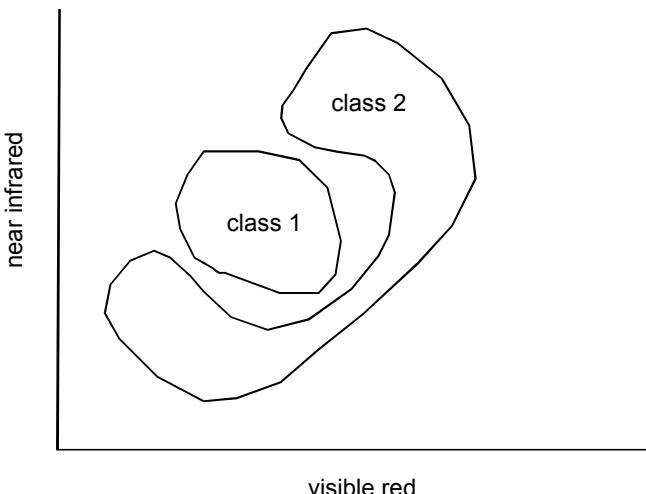
### Non-parametric Discriminant Analysis and Decision Boundary Feature Extraction (DBFE)

As with canonical analysis the application of DAFE requires good estimates of the relevant co-variance matrices. In the case of the within-class matrices, that could be difficult when the dimensionality of the data is high, as with hyperspectral imagery, but is often acceptable with data of multispectral order (i.e. several wavebands).

Even when the dimensionality of the data is acceptable, though, there is still an assumption that the various spectral classes are in some sense “clusters” of similar pixel vectors so that within class-covariance adequately describes how they spread about their mean positions, and that between-class covariance, computed from the means, is also meaningful. Provided spectral classes have been appropriately delineated beforehand, that should not present much of a concern in image labelling since, by definition, we try to segment the available data into groups whose properties match the supervised classification algorithm to be employed.

If, however, one or more of the classes were unusual in distribution, such as an (unresolved spectral) class that might encompass a range of dark and light soil types and a separate class of vegetative stubble, recorded in the visible and near infrared region, then feature reduction methods that depend on class means and covariance matrices may not work well. A class distribution such as that depicted in Fig. 10.11 is an example.

Should those types of class be suspected then it is better to avoid separability criteria that depend on class statistics and instead try to find a method that is non-parametric. Non-parametric Discriminant Analysis (NDA) (Fukunaga, 1990), and



**Fig. 10.11.** Situation in which DAFE would not be expected to perform well since the mean of class 2 would be little different from that of class 1, and the within class covariance matrix of class 2 would not reflect the actual scatter of the data

its extension to Decision Boundary Feature Extraction (Landgrebe, 2003) is such an approach. Rather than use class-based measures of mean and covariance, it uses local statistical properties, in the following manner.

In its simplest form NDA examines the relationship between the training pixels from one class (in a two class example) and their nearest neighbour training pixels from the other class. For example, let  $\mathbf{x}_{j \in s, NN_i \in r}$  be the pixel ( $j$ ) from class  $s$  that is the nearest neighbour of the  $i$ th pixel from class  $r$ :  $\mathbf{x}_{j \in s, NN_i \in r}$  is going to take the role of the mean vector in the usual type of covariance calculation as far as pixels from class  $r$  are concerned; in this case however the “mean” is different for each pixel of class  $r$  (it is its nearest neighbour – which has to be computed). We can describe the distribution of the class  $r$  pixels with respect to their nearest neighbours by a covariance like calculation. However, because we are now not describing the distribution of pixels around a class mean (a parametric description), it is better to talk about the scatter of pixels with respect to each other, and thus use the term *scatter matrix* to describe the measure.

For example, the scatter of (all of) the training pixels from class  $r$  about their nearest neighbours from class  $s$ ,  $\mathbf{x}_{j \in s, NN_i \in r}$ , is

$$S_{b1} = \mathbb{E}\{(\mathbf{x}_{i \in r} - \mathbf{x}_{j \in s, NN_i \in r})(\mathbf{x}_{i \in r} - \mathbf{x}_{j \in s, NN_i \in r})^t | \omega_r\}$$

where  $\mathbf{x}_{i \in r}$  is the  $i$ th pixel from class  $r$ ,  $\mathbb{E}$  is the expectation operator and the  $|\omega_r$  conditionality reminds us that the calculation is determined by the training pixels in class  $r$ .

We perform a similar calculation for the scatter of the training pixels from class  $s$  about their class  $r$  nearest neighbours, and then average the two measures – usually weighted by the prior probabilities (or relative training sample abundances) of the classes:

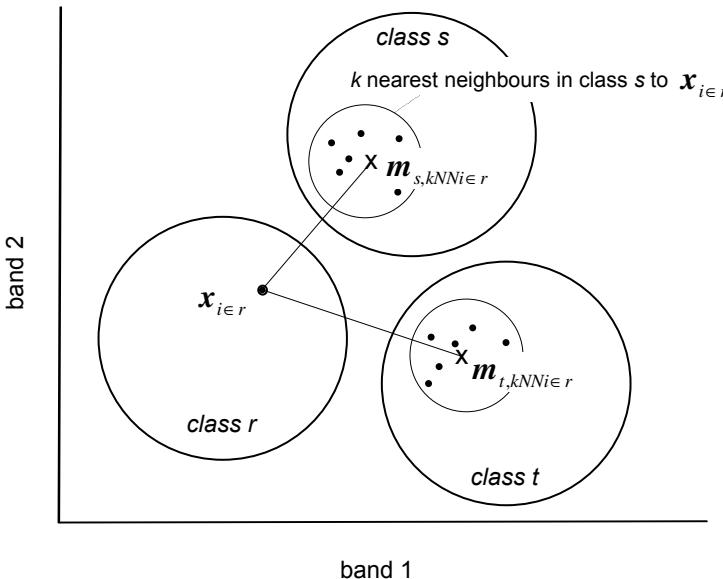
$$\begin{aligned} S_b &= S_{b1} + S_{b2} \\ &= p(\omega_r) \mathbb{E}\{(\mathbf{x}_{i \in r} - \mathbf{x}_{j \in s, NN_i \in r})(\mathbf{x}_{i \in r} - \mathbf{x}_{j \in s, NN_i \in r})^t | \omega_r\} \\ &\quad + p(\omega_s) \mathbb{E}\{(\mathbf{x}_{j \in s} - \mathbf{x}_{i \in r, NN_j \in s})(\mathbf{x}_{j \in s} - \mathbf{x}_{i \in r, NN_j \in s})^t | \omega_s\} \end{aligned}$$

More often than not NDA uses *not* just the nearest neighbour in these calculations, but instead defines a nearest neighbourhood of  $k$  class  $s$  training pixels to each class  $r$  training pixel, and then uses the local mean over that neighbourhood in the calculation of the between class scattering matrix. Let  $\mathbf{x}_{l \in s, kNN_i \in r}$  denote the  $l$ th of the  $k$  nearest neighbours from class  $s$  of pixel  $i$  from class  $r$ . Then the local (class  $s$ ) mean is defined as

$$\mathbf{m}_{s, kNN_i \in r} = \frac{1}{k} \sum_{l=1}^k \mathbf{x}_{l \in s, kNN_i \in r} \tag{10.22}$$

in which case the expression for the between class scattering matrix becomes

$$\begin{aligned} S_b &= p(\omega_r) \mathbb{E}\{(\mathbf{x}_{i \in r} - \mathbf{m}_{s, kNN_i \in r})(\mathbf{x}_{i \in r} - \mathbf{m}_{s, kNN_i \in r})^t | \omega_r\} \\ &\quad + p(\omega_s) \mathbb{E}\{(\mathbf{x}_{j \in s} - \mathbf{m}_{r, NN_j \in s})(\mathbf{x}_{j \in s} - \mathbf{m}_{r, NN_j \in s})^t | \omega_s\} \end{aligned} \tag{10.23}$$



**Fig. 10.12.** The  $k$  nearest neighbours of the  $i$ th pixel from class  $r$  in each of two other classes.

Note from (10.22) that if  $k$ , the size of the neighbourhood, is the same as the total number of training pixels available in class  $s$  then the “local” mean becomes the class mean, and the between-class scatter matrices do indeed look like covariance matrices, although taken around the mean of the opposite class rather than the mean of their own class.

Generalisation of (10.23) requires a little thought because there are as many weighted means of the pixels “from the other class” as there are “other classes.” This is illustrated in Fig. 10.12 for the case of three classes:  $r$ ,  $s$  and  $t$ . It is therefore easier to express the expectations in (10.23) in algebraic form so that for  $C$  total classes

$$S_b = \sum_{r=1}^C p(\omega_r) \sum_{c=1, c \neq r}^C \frac{1}{N_r} \sum_{i=1}^{N_r} (\mathbf{x}_{i \in r} - \mathbf{m}_{c, kNNi \in r}) (\mathbf{x}_{i \in r} - \mathbf{m}_{c, kNNi \in r})^t \quad (10.24)$$

in which the inner sum computes the expected scatter between the  $N_r$  training pixels from class  $r$  and the mean of the nearest neighbours in class  $c$  (different for each training pixel), the middle sum then changes the class ( $c$ ), still relating to the training pixels from class  $r$ , and the outer sum changes the class ( $r$ ) for which the training pixels are being considered; the latter computation is weighted by the prior probability for the class.

Having determined a non-parametric expression for among-class scatter we now need to consider the within-class scatter properties, in order to be able to use a criterion such as that in (10.19) to guide feature reduction. Fukunaga suggests using the usual form of the within-class scatter matrix (10.11), although with a data transfor-

mation that maps it to the identity matrix. Based on this assumption he shows that the NDA transformation that ranks the transformed features by decreasing value in separability is

$$\mathbf{z} = \Psi^t \Lambda^{-1/2} \Phi^t \mathbf{x}$$

where  $\Psi$  is the matrix of eigenvectors of  $S_b$ , and  $\Lambda$  is the diagonal eigenvalue matrix and  $\Phi$  is the eigenvector matrix of the within-class scatter matrix.

An alternative non-parametric expression for the within-class scatter has been proposed by Kuo and Landgrebe (2004), based on the local neighbourhood concept above in which the mean of the  $k$  neighbours from class  $r$  of the  $i$ th pixel also from class  $r$  is

$$\mathbf{m}_{r,kNNi \in r} = \frac{1}{k} \sum_{l=1,l \neq i}^k \mathbf{x}_{l \in r, kNNi \in r}$$

so that the within-class scatter matrix in the two class case is

$$\begin{aligned} S_w = & p(\omega_r) \mathbb{E} \left\{ (\mathbf{x}_{i \in r} - \mathbf{m}_{r,kNNi \in r})(\mathbf{x}_{i \in r} - \mathbf{m}_{r,kNNi \in r})^t | \omega_r \right\} \\ & + p(\omega_s) \mathbb{E} \left\{ (\mathbf{x}_{j \in s} - \mathbf{m}_{s,kNNj \in s})(\mathbf{x}_{j \in s} - \mathbf{m}_{s,kNNj \in s})^t | \omega_s \right\}. \end{aligned}$$

The procedures of Sect. 10.4.3 can then be used to find the required transformation.

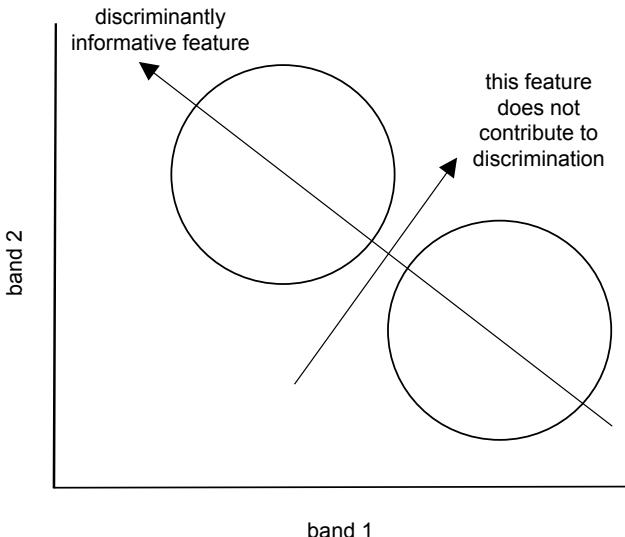
It is clear that this process might lead to a better outcome if only those training pixels in the vicinity of the decision boundaries were used in the computation of the scatter matrices. Accordingly, the calculations can be weighted to lessen the influence of neighbours further away from each other.

There are several limitations with the NDA approach, including the need both to specify the size of the neighbourhoods and the need to identify the neighbours to be used in each calculation. In contrast, even though canonical analysis is parametric in its basis, the computational demand is relatively straightforward.

Another feature reduction procedure that uses training pixels only in the vicinity of the decision boundary is Decision Boundary Feature Extraction (DBFE), summarised by Landgrebe (2003).

Briefly, it is based on the notion that (transformed) feature vectors normal to decision boundaries are discriminantly informative, whereas feature vectors that lie parallel to a decision boundary do not aid in class separation. Figure 10.13 illustrates this point with a two dimensional, two class example. The problem, therefore, is to find an effective representation of the normals to the portion of the decision boundary that is used in discrimination (even though a decision boundary may be infinite in extent theoretically, in practice only that portion in the vicinity of the training data is significant).

DBFE is a parametric procedure. It commences by estimating the class conditional means and covariance matrices, which are used to define the actual decision surface and then to classify the training pixels. Outlying pixels from each class are then removed using a Chi-squared test (see Sect. 8.2.5), and a sample of each class in the vicinity of the decision surface is selected by applying the Chi-squared test to the pixels of the opposite class, using the statistics of the first class.



**Fig. 10.13.** Transformed axes in which one of the new features is of value in separating the classes shown. The other feature, being parallel to a likely decision boundary or separating surface does not assist in discrimination.

From the sample identified, the decision surface normals are estimated in the vicinity of the training data, from which an effective decision boundary feature matrix is computed. While the dimensionality of the matrix will be the same as the original feature space, its rank may be smaller, indicating the (reduced) number of discriminantly informative features.

DBFE has a number of drawbacks, including, again, the large number of calculations required and the need to obtain reliable estimates of the original class signatures and the decision boundary feature matrix. Those parameter estimates are not reliable if the dimensionality is high and the number of training samples (especially in the vicinity of the decision surface) is limited.

#### 10.4.5

#### Non-parametric Weighted Feature Extraction (NWFE)

NWFE is a variation on the weighted version of DAFE above. It weights the samples used in the calculations of the local means and uses slightly different definitions of the among-class and within-class scattering matrices.

First, consider the calculation of a local mean for class  $r$  pixels in the vicinity of pixel  $i$  from that same class. Rather than using a set of  $k$  nearest neighbours, all the training pixels are used, but their influence on the computed value of the mean is diminished the further they are from  $\mathbf{x}_{i \in r}$ . Thus the weighted  $r$  class mean about

the  $i$ th pixel from class  $r$  is

$$\mathbf{m}_{r,i \in r} = \sum_{l=1}^{N_r} w_{l \in r, i \in r} \mathbf{x}_{l \in r}$$

where  $N_r$  is the number of training pixels in class  $r$ , and the weight  $w_{l \in r, i \in r}$  is defined by

$$w_{l \in r, i \in r} = \frac{d^{-1}(\mathbf{x}_{i \in r}, \mathbf{x}_{l \in r})}{\sum_{l=1}^{N_r} d^{-1}(\mathbf{x}_{i \in r}, \mathbf{x}_{l \in r})}$$

in which  $d^{-1}$  is the inverse of the distance between the pixel vectors in its argument. In a similar manner, the “local” mean of the class  $s$  pixels as far as the  $i$ th pixel from class  $r$ ,  $\mathbf{x}_{i \in r}$ , is concerned is

$$\mathbf{m}_{s,i \in r} = \sum_{l=1}^{N_s} w_{l \in s, i \in r} \mathbf{x}_{l \in s}$$

where the weight now is

$$w_{l \in s, i \in r} = \frac{d^{-1}(\mathbf{x}_{i \in r}, \mathbf{x}_{l \in s})}{\sum_{l=1}^{N_s} d^{-1}(\mathbf{x}_{i \in r}, \mathbf{x}_{l \in s})}$$

Using these new definitions of the means, the among class and within class scatter matrices are now computed, for the multiclass case, as

$$S_b = \sum_{r=1}^C p(\omega_r) \sum_{c=1, c \neq r}^C \frac{1}{N_r} \sum_{i=1}^{N_r} w_{i \in r, c} (\mathbf{x}_{i \in r} - \mathbf{m}_{c,i \in r})(\mathbf{x}_{i \in r} - \mathbf{m}_{c,i \in r})^t \quad (10.25a)$$

$$S_w = \sum_{r=1}^C p(\omega_r) \frac{1}{N_r} \sum_{i=1}^{N_r} w_{i \in r, r} (\mathbf{x}_{i \in r} - \mathbf{m}_{r,i \in r})(\mathbf{x}_{i \in r} - \mathbf{m}_{r,i \in r})^t \quad (10.25b)$$

where the weights are defined by

$$w_{i \in r, \xi} = \frac{d^{-1}(x_{i \in r}, m_{\xi, i \in r})}{\sum_{l=1}^{N_r} d^{-1}(x_{i \in r}, m_{\xi, i \in r})}$$

with  $\xi = c$  or  $r$ , in (10.25a) and (10.25b) respectively.

To avoid problems with reliable estimation, or even singularity, the within-class scatter matrix of (10.25b) is sometimes replaced by an approximate form (see Sect. 13.7). In particular, Kuo and Landgrebe (2004) use

$$S'_w = 0.5 S_w + 0.5 \operatorname{diag} S_w$$

Having established the form of the among-class and within-class scatter matrices, the required features can be found by using the eigenvectors corresponding to the largest eigenvalues of

$$J = \Sigma_{w,y}^{-1} \Sigma_{A,y}$$

With the newly defined scatter matrices, the use of this criterion for finding the transformation that gives best separability is tantamount to using (10.17).

### 10.4.6 Arithmetic Transformations

Depending upon the application, feature reduction prior to classification can sometimes be carried out using simpler arithmetic operations than the transformations treated in the foregoing sections. As an illustration, taking the differences of multispectral imagery from different dates can yield difference data that can be processed for change, by comparison with the need to classify *all* the data if the preprocessing step is not adopted.

A second example is the use of the simple ratio of infrared to visible data as a vegetation index. This allows vegetation classification to be performed on the ratio data alone. More sophisticated vegetation indices exist and these can be considered as data reduction transformations. The most commonly encountered are the following in which the bands designated are those from the multispectral scanners on Landsats 1–3. The band numbers need to be redefined to refer to Landsats 4 onwards, and other sensors.

$$\text{VI} = (\text{band 7} - \text{band 5}) / (\text{band 7} + \text{band 5}) \quad (\text{vegetation index})$$

$$\text{TVI} = \sqrt{\text{VI} + 0.5} \quad (\text{transformed vegetation index})$$

These and others are discussed in Myers (1983).

## References for Chapter 10

For more mathematical details on measures of divergence and Jeffries-Matusita distance the reader is referred to Duda, Hart and Stork (2001) and Kailath (1967). A detailed discussion on transformed divergence will be found in Swain and King (1973).

A good introductory discussion on canonical analysis in remote sensing is given in the paper by Jensen and Waltz (1979). More detailed descriptions from a remote sensing viewpoint have been given by Merembeck et al. (1977), Merembeck and Turner (1980) and Eppler (1976). These also contain results that illustrate its suitability as a feature reduction tool.

Detailed mathematical descriptions of canonical analysis, as a statistical tool, will be found in Seal (1964) and Tatsuoka (1971). Seal in particular gives the results of hand calculations on two and three dimensional data sets.

N.A. Campbell and W.R. Atchley, 1981: The Geometry of Canonical Variate Analysis. Systematic Zoology, 30, 268–280.

- R.O. Duda, P.E. Hart and D.G. Stork, 2001: Pattern Classification, 2e, N.Y., Wiley.
- W. Eppler, 1976: Canonical Analysis for Increased Classification Speed and Channel Selection, IEEE Trans. Geoscience Electronics, GE-14, 26–33.
- K. Fukunaga, 1990: Introduction to Statistical Pattern Recognition, London, Academic.
- S.K. Jensen and F.A. Waltz, 1979: Principal Components Analysis and Canonical Analysis in Remote Sensing, Proc. American Soc. of Photogrammetry 45th Ann. Meeting, 337–348.
- T. Kailath, 1967: The Divergence and Bhattacharyya Distance Measures in Signal Selection, IEEE Trans. Communication Theory, COM-15, 52–60.
- B.-C. Kuo and D.A. Landgrebe, 2004: Nonparametric Weighted Feature Extraction for Classification. IEEE Trans. Geoscience and Remote Sensing, 42, 1096–1105.
- D.A. Landgrebe, 2003: Signal Theory Methods in Multispectral Remote Sensing, N.Y., Wiley.
- P.W. Mausel, W.J. Kramber and J.K. Lee, 1990: Optimum Band Selection for Supervised Classification of Multispectral Data. Photogrammetric Engineering and Remote Sensing, 56, 55–60.
- B.F. Merembeck, F.Y. Borden, M.H. Podwysocki and D.N. Applegate, 1976: Application of Canonical Analysis to Multispectral Scanner Data, Proc. 14th Symp. Applications of Computer Methods in the Mineral Industries, 867–879.
- B.F. Merembeck and B.J. Turner, 1980: Directed Canonical Analysis and the Performance of Classifiers under its Associated Linear Transformation, IEEE Trans. Geoscience and Remote Sensing, GE-18, 190–196.
- V.I. Myers, 1983: Remote Sensing Applications in Agriculture. In: R.N. Colwell (Ed.): Manual of Remote Sensing, 2e, American Soc. of Photogrammetry, Falls Church.
- H. Seal, 1964: Multivariate Statistical Analysis for Biologists. London, Methuen.
- P.H. Swain and S.M. Davis, (Eds.), 1978: Remote Sensing: The Quantitative Approach, N.Y., McGraw-Hill.
- P.H. Swain and R.C. King, 1973: Two Effective Feature Selection Criteria for Multispectral Remote Sensing, Proc. First Int. Joint Conf. on Pattern Recognition, 536–540, November.
- P.H. Swain, T.V. Robertson and A.G. Wacker, 1971: Comparison of Divergence and B-Distance in Feature Selection, Information Note 020871, Laboratory for Applications of Remote Sensing, Purdue University, West Lafayette.
- M.M. Tatsuoka, 1971: Multivariate Analysis, N.Y., Wiley.
- A.G. Wacker, 1971: The Minimum Distance Approach to Classification, Ph.D. Thesis, Purdue University, West Lafayette.

## Problems

**10.1** Kailath (1967) shows that the probability of making an error in labelling a pattern as belonging to one of two classes with equal prior probabilities is bounded according to

$$\frac{1}{16}(2 - J_{ij})^2 \leq P_E \leq \frac{1}{4}(2 - J_{ij})$$

where  $J_{ij}$  is the Jeffries-Matusita distance between the classes. Determine and plot the upper and lower bounds on classification accuracy for a two class problem, as a function of  $J_{ij}$ . You may wish to compare this to an empirical relationship between classification accuracy and  $J_{ij}$  found by Swain and King (1973).

**10.2** Consider the training data given in problem 8.1. Suppose it is required to use only one feature to characterise each spectral class. By computing pairwise transformed divergence measures ascertain the best feature to retain if:

- (a) only classes 1 and 2 are to be considered
- (b) only classes 2 and 3 are to be considered
- (c) all three classes are to be considered.

In each case estimate the maximum possible classification accuracy.

**10.3** Using the same data as in problem 10.2, perform feature reductions if possible using principal component transformations if the covariance matrix is generated using

- (a) only classes 1 and 2
- (b) only classes 2 and 3
- (c) all three classes.

**10.4** Using the same data as in problem 10.2, compute a canonical analysis transformation for all three classes of data and see whether the classes have better discrimination in the transformed axes.

**10.5** Suppose the mean vectors and covariance matrices have been determined, using training data, for a particular image of an agricultural region. Because of the nature of the land use, the region consists predominantly of fields that are large compared with the effective ground dimensions of a pixel, and within each field there is a degree of similarity among the pixels, owing to its use for a single crop type.

Suppose you delineate a field from the rest of the image (either manually or automatically) and then compute the mean vector and covariance matrix for all the pixels in that field. Describe how pairwise divergence, or Jeffries-Matusita distance could be used to classify the complete *field* of pixels into one of the training classes.

**10.6** The application of rotational transforms such as principal components and canonical analysis cannot improve intrinsic separability – i.e. the separability possible in the original data with all dimensions retained. Why?

**10.7** The principal components transformation can be used for feature selection. What advantages and disadvantages does it have compared with canonical analysis?

**10.8** Two classes have the statistics:

$$\begin{aligned} \mathbf{m}_1 &= \begin{bmatrix} 10 \\ 20 \end{bmatrix} & \Sigma_1 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \\ \mathbf{m}_2 &= \begin{bmatrix} 10 \\ 20 \end{bmatrix} & \Sigma_2 &= \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \end{aligned}$$

- (a) Can a minimum distance classifier work for this data?
- (b) Calculate the JM distance between the classes. Are they separable?
- (c) Assuming equal prior probabilities, classify the pixel vector  $\mathbf{x} = [12 \ 30]^t$ .

**10.9** Both training and testing data are required for developing a Gaussian maximum likelihood classification. What reason might there be for low classification accuracy on the training data? If the classification accuracy is high on the training data but low on the testing data, what could be the reason?

# 11

## Image Classification Methodologies

### 11.1

#### Introduction

In principle, classification of multispectral image data should be straightforward. However to achieve results of acceptable accuracy care is required first in choosing the analytical tools to be used and then in applying them. In the following the classical analytical procedures of supervised and unsupervised classification are examined from an operational point of view, with their strengths and weaknesses highlighted. These approaches are often acceptable; however more often a judicious combination of the two will be necessary to attain optimal results. A hybrid supervised/unsupervised strategy is therefore also presented.

Other compound classification approaches are also possible including the hierarchical decision tree methods covered in Sect. 11.8.

### 11.2

#### Supervised Classification

##### 11.2.1

###### Outline

As discussed in Chap. 8 the underlying requirement of supervised classification techniques is that the analyst has available sufficient known pixels for each class of interest that representative signatures can be developed for those classes. These prototype pixels are often referred to as training data, and collections of them, identified in an image and used to generate class signatures, are called training fields. The step of determining class signatures is frequently called training.

Particular care needs to be taken when attempting to generate signatures for hyperspectral data sets. As a result, procedures for classifying hyperspectral image data

are treated separately in Chap. 13. Nevertheless, it is possible to condition hyperspectral data (for example, through feature selection) so that the material outlined here is still relevant.

Signatures generated from the training data will be of a different form depending on the classifier type to be used. For parallelepiped classification the class signatures will be the upper and lower bounds of brightness in each spectral band. For minimum distance classification the signatures will be the mean vectors of the training data for each class, while for maximum likelihood classification both class mean vectors and covariance matrices constitute the signatures. For neural network and support sector machine classifiers the collection of weights define the boundaries between classes. While they do not represent class signatures as such they are the inherent properties of the classifier, learnt from training data, that allow classes to be discriminated.

By having the labelled training data available beforehand, from which the signatures are estimated, the analyst is, in a relative sense, teaching the classification algorithm to recognise the spectral characteristics of each class, thereby leading to the term *supervised* as a qualification relating to the algorithm's learning about the data with which it has to work.

As a proportion of the full image to be analysed the amount of training data would represent less than 1% to 5% of the pixels. The learning phase therefore, in which the analyst plays an important part in the *a priori* labelling of pixels, is performed on a very small part of the image. Once trained, the classifier is then asked to attach labels to *all* the image pixels by using the class estimates provided to it.

The steps in this fundamental outline are now examined in more detail, noting the practical issues that should be considered to achieve reliable results.

## **11.2.2**

### **Determination of Training Data**

The major step in straightforward supervised classification is the prior identification of training pixels. This may involve the expensive enterprise of field visits, or may require use of reference data such as topographic maps and air photographs. In the latter, a skilled photointerpreter may be required to determine the training data. Once training fields are suitably chosen they have to be related to the pixel addresses in the satellite imagery. Sometimes training data can be chosen by photointerpretation from image products formed from the multispectral data to be classified. Generally however this is restricted to major cover types and again can require a great deal of photointerpretive skill if more than a simple segmentation of the image is required.

Some image processing systems have digitizing tables that allow map data – such as polygons of training pixels, i.e. training fields – to be taken from maps and superimposed over the image data. While this requires a registration of the map and image, using the procedures of Sect. 2.4, it represents an unbiased method for choosing the training data. It is important however, as with all training procedures based upon field or reference data, that the training data be recorded at about the same time as the multispectral data to be classified. Otherwise errors resulting from temporal variations may arise.

It is necessary to identify training data at least for all classes of interest and preferably for all apparent classes in the segment of image to be analysed. In either case, and particularly if the selection of training data is not exhaustive or representative, it is prudent to use some form of threshold or limit if the classification is of the minimum distance or maximum likelihood variety; this will ensure poorly characterised pixels are not erroneously labelled. Limits in minimum distance classification can be imposed by only allowing a pixel to be classified if it is within a prespecified number of standard deviations of the nearest mean. For maximum likelihood classification a limit may be applied by the use of thresholds on the discriminant functions. Having so limited a classification, pixels in the image which are not well represented in the training data will not be classified. This will identify weaknesses in the selection of the training sets which can then be rectified and the image re-classified. Repeated refinement of the training data and reclassification in this manner can be carried out using a representative portion of the image data.

### **11.2.3 Feature Selection**

The cost of the classification of a full image segment is reduced if bands or features that do not aid discrimination significantly are removed. After training is complete feature selection can be carried out using the separability measures presented in Chap. 10. The recommended measures are transformed divergence, if maximum likelihood signatures have been generated, or Euclidean distance if the signatures have been prepared for minimum distance classification.

Separability measures can also be used to assess whether any pair of classes are so similar in multispectral space that significant misclassification will occur if they are both used. Should such a pair be found the analyst should give consideration to merging them to form a single class.

If hyperspectral data is being considered feature selection can be a crucial step. Yet, unfortunately, many separability measures used to effect feature selection are themselves dependent on class covariance matrices. The material in Sect. 13.7 is then particularly relevant.

### **11.2.4 Detecting Multimodal Distributions**

The most common algorithm for supervised classification is that based upon maximum likelihood estimation of class membership of an unknown pixel using multivariate normal distribution models for the classes. Its attraction lies in its ability to model class distributions that are elongated to different extents in different directions in multispectral space and its consequent theoretical guarantee that, if properly applied, it will lead to minimum average classification error. However, its major limitation in this regard is that the classes must be representable as multivariate normal distributions. Often the information classes of interest will not appear as single

distributions but rather are best resolved into a set of constituent spectral classes or sub-classes. Should these spectral classes not be properly identified beforehand, the accuracy of supervised maximum likelihood classification will suffer. Multimodal classes can be identified to an extent using clustering algorithms; indeed this is the basis of the hybrid classification methodology developed in Sect. 11.4 below. A simple, yet rather more limited means, by which multimodal behaviour can be assessed is to examine scatterplots of the data in each training class. A scatterplot is a two dimensional multispectral space with user defined axes. An infrared versus visible red scatterplot for “vegetation” prototype pixels could show, for example, two distinct regions of data concentration, corresponding to sub-classes of “grassland” and “trees”.

Should any of the sets of training data be found to be multimodal, steps should be taken to resolve them into the appropriate sub-classes in order to minimise classification error. Again clustering of the training sets could be used to do this, although it is frequently straightforward to identify groups of image pixels corresponding to each of the data modes in a scatterplot, thereby allowing the analyst to subdivide the corresponding training fields.

### **11.2.5 Presentation of Results**

Two types of output are available from a classification. One is the thematic (or class) map in which pixels are given a label (represented by a colour or symbol) to identify them with a class. The other output is a table that summarises the number of pixels in the image found to belong to each class. The table can be interpreted also as a table of areas, in hectares. However that requires either that the user has resampled the image data to a map grid beforehand, so that the pixels correspond to an actual area on the ground, or that the user takes account of any systematic pixel overlap such as the 23 m overlap of Landsat MSS pixels caused by the detector sampling strategy (see Appendix A). In that case it is important to recall that the effective MSS pixel is 56 m × 79 m and thus represents an area of 0.4424 ha for Landsats 1 to 3.

### **11.2.6 Effect of Resampling on Classification**

The utility of remote sensing image data is improved if it is registered to a map base. As discussed in Sect. 2.4.1.3 several interpolation techniques can be used to synthesise pixel values on the map grid, the most common being nearest neighbour resampling and resampling by cubic convolution. In the former, original image pixels are simply relocated onto a geometrically correct map grid whereas in the latter new pixel brightness values are synthesised by interpolating over a group of sixteen pixels.

Usually it is desirable to have the thematic maps produced by classification registered to a map base. This can be done either by rectifying the image before classification or by rectifying the actual thematic map (in which case nearest neighbour

resampling is the only option). An advantage in correcting the image beforehand is that it is often easier to relate reference data and ground truth information to the image if it is in correct geometric registration to a map. However a drawback with doing this from a data analysis/information extraction point of view is that the data is then processed before classification is attempted. That preprocessing could add noise and uncertainty to the pixel brightness values and therefore prejudice subsequent classification accuracy. Accordingly, a good rule wherever possible is not to correct the data before classification. Should it be necessary to rectify the data then nearest neighbour interpolation should be used in the resampling stage if possible.

The influence of resampling on classification has been addressed by Billingsley (1982), Verdin (1983) and Forster and Trinder (1984) who show examples of how cubic convolution interpolation can have a major influence across boundaries such as that between vegetation and water, leading to uncertainties in classification.

When images in a multitemporal sequence have to be classified to extract change information it is necessary to perform image to image registration (which could alternatively consist of registering all the images to a reference map). Since registration cannot be avoided in this case, nearest neighbour resampling should be used.

## 11.3 Unsupervised Classification

### 11.3.1 Outline, and Comparison with Supervised Methods

Unsupervised classification is an analytical procedure based on clustering, using algorithms such as those described in Chap. 9. Application of clustering partitions the image data in multispectral space into a number of spectral classes, and then labels all pixels of interest as belonging to one of those spectral classes, although the labels are purely symbolic (e.g. A, B, C, . . . , or class 1, class 2, . . . ) and are as yet unrelated to ground cover types. Hopefully the classes will be unimodal; however, if simple unsupervised classification is of interest, this is not essential.

Following segmentation of the multispectral space by clustering, the clusters or spectral classes are associated with information classes – i.e. ground cover types – by the analyst. This *a posteriori* identification may need to be performed explicitly only for classes of interest. The other classes will have been used by the algorithm to ensure good discrimination but will remain labelled only by arbitrary symbols rather than by class names.

The identification of classes of interest against reference data is often more easily carried out when the spatial distribution of spectrally similar pixels has been established in the image data. This is an advantage of unsupervised classification and the technique is therefore a convenient means by which to generate signatures for spatially elongated classes such as rivers and roads.

In contrast to the *apriori* use of analyst-provided information in supervised classification, unsupervised classification is a segmentation of the data space in the absence

of any information provided by the analyst. Analyst information is used only to attach information class (or ground cover type, or map) labels to the segments established by clustering. Clearly this is an advantage of the approach. However it is a time-consuming procedure computationally by comparison to techniques for supervised classification. This can be demonstrated by comparing, for example, multiplication requirements of the iterative clustering algorithm of Sect. 9.3 with the maximum likelihood classification decision rule of Sect. 8.2.3.

Suppose a particular classification exercise involves  $N$  spectral bands and  $C$  classes. Maximum likelihood classification requires  $CPN(N + 1)$  multiplications where  $P$  is the number of pixels in the image segment of interest. By comparison, clustering of the data requires  $PCI$  distance measures for  $I$  iterations. Each distance calculation demands  $N$  multiplications<sup>1</sup>, so that the total number of multiplications for clustering is  $PCIN$ . Thus the speed comparison of the two approaches is approximately  $(N + 1)/I$  for maximum likelihood classification compared with clustering. For Landsat MSS data, therefore, in a situation where all 4 spectral bands are used, clustering would have to be completed within 5 iterations to be speed competitive with maximum likelihood classification. Frequently 20 times this number of iterations is necessary to achieve an acceptable clustering. Training the classifier would add about a 10% loading to its time demand; however a significant time loading should also be added to clustering to account for the labelling phase. Often this is done by associating pixels with the nearest (Euclidean distance) cluster. However, sometimes Mahalanobis or maximum likelihood distance labelling is used. This adds substantially to the cost of clustering.

Because of the time demand of clustering algorithms, unsupervised classification is often carried out with small image sequents. Alternatively a representative subset of data is used in the actual clustering phase in order to cluster or segment the multispectral space. That information is then used to assign all the image pixels to a cluster.

When comparing the time requirements of supervised and unsupervised classification it must be recalled that a large demand on user time is required in training a supervised procedure. This is necessary both for determining training data and then identifying training pixels by reference to that data. The corresponding step in unsupervised classification is the *a posteriori* labelling of clusters. While this still requires user effort in determining labelled prototype data, not as much may be required. As noted earlier, data is only required for those classes of interest; moreover only a handful of labelled pixels is necessary to identify a class. By comparison, sufficient training pixels per class are required in supervised training to ensure reliable estimates of class signatures are generated.

A final point that must be taken into account when contemplating unsupervised classification via clustering is that there is no facility for including prior probabilities of class membership. By comparison the decision functions for maximum likelihood classification can be biased by previous knowledge or estimates of class membership.

---

<sup>1</sup> Usually distance squared is calculated avoiding the need to evaluate the square root operation in (9.1).

### 11.3.2

#### Feature Selection

Most clustering procedures used for unsupervised classification in remote sensing generate the mean vector and covariance matrix for each cluster found. Accordingly separability measures can be used to assess whether feature reduction is necessary or whether some clusters are sufficiently similar spectrally that they should be merged. These are only considerations of course if the clustering is generated on a sample of data, with a second phase used to allocate all image pixels to a cluster. Feature selection would be performed between the two phases.

## 11.4

### A Hybrid Supervised/Unsupervised Methodology

#### 11.4.1

#### The Essential Steps

The strength of supervised classification based on the maximum likelihood procedure is that it minimises classification error for classes that are distributed in a multivariate normal fashion. Moreover, it can label data relatively quickly. Its major drawback lies in the need to have delineated unimodal spectral classes beforehand. This, however, is a task that can be handled using clustering, using a representative subset of image data. Used for this task, unsupervised classification performs the valuable function of identifying the existence of all spectral classes, yet it is not expected to perform the entire classification. Consequently, the rather logical hybrid classification procedure outlined below can be envisaged. This is due to Fleming et al. (1975).

- Step 1: Use Clustering to determine the spectral classes into which the image resolves. For reasons of economy this is performed on a representative subset of data. Spectral class statistics are also produced from this unsupervised step.
- Step 2: Using available ground truth or other reference data associate the spectral classes (or clusters) with information classes (ground cover types). Frequently, there will be more than one spectral class for each information class.
- Step 3: Perform a feature selection evaluation to see whether all features (bands) need to be retained for reliable classification.
- Step 4: Using the maximum likelihood algorithm, classify the entire image into the set of spectral classes.
- Step 5: Label each pixel in the classification by the ground cover type associated with each spectral class.

It is now instructive to consider some of these steps in detail and thereby introduce some useful practical concepts. The method depends for its accuracy (as do all classifications) upon the skills and experience of the analyst. Consequently, it is

not unusual in practice to iterate over sets of steps as experience is gained with the particular problem at hand.

### 11.4.2

#### Choice of the Clustering Regions

Clustering is employed in Step 1 above to determine the spectral classes, using a subset of the image data. It is recommended that about 3 to 6 small regions, or so-called candidate clustering areas, be chosen for this purpose. These should be well spaced over the image and located such that each one contains several of the cover types (information classes) of interest and such that all cover types are represented in the collection of clustering areas. An advantage in choosing heterogeneous regions to cluster, as against apparently homogeneous training areas used in supervised classification, is that mixture pixels lying on class boundaries will be identified as legitimate spectral classes.

If an iterative clustering procedure is used, the analyst will have to prespecify the number of clusters expected in each candidate area. Experience has shown that, on the average, there are about 2 to 3 spectral classes per information class. This number should be chosen, with a view to removing or rationalising unnecessary clusters at a later stage.

It is of value to cluster each region separately as this saves computation, and produces cluster maps within those areas with more distinct class boundaries than would be the case if all regions were pooled beforehand.

### 11.4.3

#### Rationalisation of the Number of Spectral Classes

When clustering is complete the spectral classes are then associated with information classes using available reference data. It is then necessary to see whether any spectral classes or clusters can be discarded, or more importantly, whether sets of clusters can be merged, thereby reducing their number and leading ultimately to a faster classification. Decisions about merging can be made on the basis of separability measures, such as those treated in Chap. 10.

During this rationalisation procedure it is useful to be able to visualise the locations of the spectral classes. For this a bispectral plot can be constructed. The bispectral plot is not unlike a two dimensional scatter plot view of the multispectral space in which the data appears. However, rather than having the individual pixels shown, the class or cluster means are located according to their spectral components. In some exercises the most significant pair of spectral bands would be chosen in order to view the relative locations of the cluster centres. These could be infrared and red bands for a vegetation study. Sometimes averages over several bands may be useful for one of the axes. In general, the choice of bands and combinations to use in a bi-spectral plot will depend on the sensor and application. Sometimes several plots with different bands will give a fuller appreciation of the distribution of classes in multispectral space.

## 11.5

### Assessment of Classification Accuracy

#### 11.5.1

##### Using a Testing Set of Pixels

At the completion of a classification exercise it is necessary to assess the accuracy of the results obtained. This will allow a degree of confidence to be attached to the results and will serve to indicate whether the analysis objectives have been achieved.

Accuracy is determined empirically, by selecting a sample (desirably an independent random sample) of pixels from the thematic map and checking their labels against classes determined from reference data (desirably gathered during site visits). Often reference data is referred to as ground truth, and the pixels selected for accuracy checking are called *testing* pixels. From these checks the percentage of pixels from each class in the image labelled correctly by the classifier can be estimated, along with the proportions of pixels from each class erroneously labelled into every other class. These results are then expressed in tabular form, often referred to as a *confusion or error matrix*, of the type illustrated in Table 11.1. The values listed in the table represent the number of ground truth pixels, in each case, correctly and incorrectly labelled by the classifier. It is common to average the percentage of correct classifications and regard this the overall classification accuracy (in this case 83%), although a better measure globally would be to weight the average according to the areas of the classes in the map.

Sometimes a distinction is made between errors of omission and errors of commission, particularly when only a small number of cover types is of interest, such as in the estimation of the area of a single crop in agricultural applications. Errors of omission correspond to those pixels belonging to the class of interest that the classifier has failed to recognise whereas errors of commission are those that correspond to pixels from other classes that the classifier has labelled as belonging to the class of interest. The former refer to columns of the confusion matrix, whereas the latter refer to rows.

When interpreting an error matrix of the type shown in Table 11.1 from the point of view of a particular class, it is important to understand that different indications

**Table 11.1.** Illustration of a confusion matrix used in assessing the accuracy of a classification

		Ground truth classes			Total
		A	B	C	
Thematic map classes	A	35	2	2	39
	B	10	37	3	50
	C	5	1	41	47
Number of ground truth pixels		50	40	46	136

of class accuracies will result according to whether the number of correct pixels for a class is divided by the total number of reference (ground truth) pixels for the class (the corresponding column sum in Table 11.1) or the total number of pixels the classifier attributes to the class (the row sum in Table 11.1). Consider class  $B$  in Table 11.1, for example. As noted, 37 of the reference data pixels have been correctly labelled. This represents  $37/40 \equiv 93\%$  of the ground truth pixels for the class. We interpret this measure, which Congalton and Green (1999) refer to as the Producer's accuracy, as the probability that the classifier has labelled the image pixel as  $B$  given that the actual (ground truth) class is  $B$ . As a user of a thematic map produced by a classifier we are more interested in the probability that the actual class is  $B$  given that the pixel has been labelled  $B$  (on the thematic map) by the classifier. This is what Congalton and Green refer to as the User accuracy, and for this example is  $37/50 \equiv 74\%$ . Thus only 74% of the pixels labelled  $B$  on the thematic map are correct, even though the classifier coped with 93% of the  $B$  class reference data. This distinction is important and leads one to believe that the User accuracy is the figure that should most often be adopted.

Some authors prefer to use the kappa coefficient as a measure of map accuracy (Hudson and Ramm 1987, Congalton and Green 1999). This is defined in terms of the elements of the error matrix; let these be represented by  $x_{ij}$ , and suppose the total number of test pixels (observations) represented in the error matrix is  $P$ . Also, let

$$x_{i+} = \sum_j x_{ij} \text{ (i.e. the sum over all columns for row } i)$$

$$x_{+j} = \sum_i x_{ij} \text{ (i.e. the sum over all rows for column } j)$$

then the kappa estimate is defined by

$$\kappa = \frac{P \sum_k x_{kk} - \sum_k x_{k+}x_{+k}}{P^2 - \sum_k x_{k+}x_{+k}}$$

Choice of the sample of pixels for accuracy assessment is an important consideration. Perhaps the simplest strategy for evaluating classifier performance is to choose a set of testing fields for each class, akin to the training fields used to estimate class signatures. These testing fields are also labelled using available reference data, presumably at the same time as the training areas. After classification the accuracy of the classifier is determined from its performance on the test pixels. Another approach, with perhaps more statistical significance since it avoids correlated near-neighbouring pixels, is to choose a random sample of individual pixels across the thematic map for comparison with reference data. A difficulty that can arise with random sampling in this manner is that it is area-weighted. That is, large classes tend to be represented by a larger number of sample points than the smaller classes; indeed some very small classes may not be represented at all. Assessment of the accuracy

of labelling small classes will therefore be prejudiced. To avoid this it is necessary to ensure small classes are represented adequately. An approach that is widely adopted is *stratified random sampling* in which the user first of all decides upon a set of strata into which the image is divided. Random sampling is then carried out within each stratum. The strata could be any convenient area segmentation of the thematic map, such as gridcells. However the most appropriate stratification to use is the actual thematic classes themselves. Consequently, the user should choose a random sample within each thematic class to assess the classification accuracy of that class.

If one adopts random sampling, stratified by class, the question that must then be answered is how many test pixels should be chosen within each class to ensure that the results entered into the confusion matrix of Table 11.1 are an accurate reflection of the performance of the classifier, and that the percentage correct classification so-derived is a reliable estimate of the real accuracy of the thematic map. To illustrate this point, a sample of one pixel from a particular class will suggest an accuracy of 0% or 100% depending on its match to ground truth. A sample of 100 pixels will clearly give a more realistic estimate. A number of authors have addressed this problem, using binomial statistics, in the following manner.

Let the pixels from a particular category in a thematic map be represented by the random variable  $x$  that takes on the value 1 if a pixel is correctly classified and 0 otherwise. Suppose the true map accuracy for that class is  $\theta$  (which is what we wish to estimate by sampling). Then the probability of  $x$  pixels being correct in a random sample of  $n$  pixels from that class is given by the binomial probability

$$p(x; n, \theta) = {}^n C_x \theta^x (1 - \theta)^{n-x} \quad x = 0, 1, \dots, n . \quad (11.1)$$

Van Genderen et al. (1978) determine the minimum sample size, by noting that if the sample is too small there is a finite chance that those pixels selected could all be labelled correctly (as for example in the extreme situation of one pixel considered above). If this occurs then a reliable estimate of the map accuracy clearly has not been obtained. Such a situation is described by  $x = n$  in (11.1), giving as the probability for all  $n$  samples being correct

$$p(n; n, \theta) = \theta^n .$$

Van Genderen et al. have evaluated this expression for a range of  $\theta$  and  $n$  and have noted that  $p(n; n, \theta)$  is unacceptably high if it is greater than 0.05 – i.e. if more than 5% of the time there is a chance of selecting a perfect sample from a population in which the accuracy is actually described by  $\theta$ . A selection of their results is given in Table 11.2. In practice, these figures should be exceeded to ensure representative outcomes are obtained. Van Genderen et al. consider an extension of the results in Table 11.2 to the case of encountering set levels of error in the sampling, from which further recommendations are made concerning desirable sample sizes.

Rosenfield et al. (1982) have also determined guidelines for selecting minimum sample sizes. Their approach is based upon determining the number of samples required to ensure that the sample mean – i.e. the number of correct classifications divided by the total number of samples per category – is within 10% of the population

**Table 11.2.** Minimum sample size necessary per category (after Van Genderen et al. 1978)

Classification accuracy	Sample size
0.95	60
0.90	30
0.85	20
0.80	15
0.60	10
0.50	5

**Table 11.3.** Minimum sample size necessary per category (after Rosenfield et al. 1982)

Classification accuracy	Sample size
0.85	19
0.80	30
0.60	60
0.50	60

mean (i.e. the map accuracy for that category) at a 95% confidence level. Again this is estimated from binomial statistics, although using the cumulative binomial distribution. Table 11.3 illustrates the results obtained; while these results agree with Table 11.2 for a map accuracy of 85% the trends about this point are opposite.

This perhaps is not surprising since the two approaches commence from different viewpoints. Rosenfield et al. are interested in ensuring that the accuracy indicated from the samples (i.e. sample mean) is a reasonable (constant) approximation of the actual map accuracy. In contrast, Van Genderen et al. base their approach on ensuring that the set of samples is representative. Both have their merits and in practice one may wish to choose a compromise of between 30 and 60 samples per category.

Once accuracy has been estimated through sampling it is important to place some confidence on the actual figures derived for each category. In fact it is useful to be able to express an interval within which the true map accuracy lies (with say 95% certainty). This interval can be determined from the accuracy estimate for a class using the expression (Freund, 1992)

$$p \left\{ -z_{\alpha/2} < \frac{x - n\theta}{\sqrt{n\theta(1 - \theta)}} < z_{\alpha/2} \right\} = 1 - \alpha \quad (11.2)$$

where  $x$  is the number of correctly labelled pixels in a sample of  $n$ ;  $\theta$  is the true map accuracy (which we currently are estimating in the usual way by  $x/n$ ) and  $1 - \alpha$  is a confidence limit. If we choose  $\alpha = 0.05$  then the above expression says that the probability that  $(x - n\theta)/\sqrt{n\theta(1 - \theta)}$  will be between  $\pm z_{\alpha/2}$  is 95%;  $\pm z_{\alpha/2}$  are points on the *normal* distribution between which  $1 - \alpha$  of the population is contained. For  $\alpha = 0.05$ , tables show  $z_{\alpha/2} = 1.960$ . Equation (11.2) is derived from properties of the normal distribution; however for a large number of samples (typically 30 or

more) the binomial distribution is adequately represented by a normal model making (11.2) acceptable. Our interest in (11.2) is seeing what limits it gives on  $\theta$ . It is shown readily, at the 95% level of confidence, that the extreme values of  $\theta$  are given by

$$\frac{x + 1.921 \pm 1.960 \{x(n-x)/n + 0.960\}^{1/2}}{n + 3.842} \quad (11.3)$$

As an illustration, suppose  $x = 294$ ,  $n = 300$  for a particular category. Then ordinarily we would use  $\bar{x} = x/n = 0.98$  as an estimate of  $\theta$ , the true map accuracy for the category. Equation (11.3) however shows, with 95% confidence, that our estimate of  $\theta$  is bounded by

$$0.9571 < \theta < 0.9908 .$$

Thus the accuracy of the category in the thematic map is somewhere between 95.7% and 99.1%.

This approach has been developed by Hord and Brunner (1976) who produced tables of the upper and lower limits on the map accuracy as a function of sample size and sample mean (or accuracy)  $\bar{x} = x/n$ .

### 11.5.2

#### The Leave One Out Method of Accuracy Assessment – Cross Validation

An interesting accuracy assessment method, which does not depend on developing a testing set of pixels, is the Leave One Out (LOO) approach. It is based on removing one of the training set of pixels, training the classifier on the remainder and using the trained classifier to label the pixel left out. That pixel is replaced and another removed and the process repeated. This is done for all pixels in the training set. The average classification accuracy is then determined. Provided the original training pixels are representative, this method produces an unbiased estimate of classification accuracy (Landgrebe, 2003).

The Leave One Out method is a special case of cross validation (Duda, Hart and Stork, 2001) in which the available labelled pixels are divided into  $k$  subsets. One of those subsets is used as the testing data and the remainder aggregated to form the training set. The process is repeated  $k$  times, so that each subset in turn is used as the testing data and the others for training.

## 11.6

### Case Study 1: Irrigated Area Determination

It is the purpose of this case study to demonstrate a simple classification, carried out using the hybrid strategy of Sect. 11.4. Rather than being based upon iterative clustering and maximum likelihood classification it makes use of a single pass clustering algorithm of the type presented in Sect. 9.6 and a minimum distance classifier as described in Sect. 8.3.

The problem presented was to use classification of Landsat Multispectral Scanner image data to assess the hectarage of cotton crops being irrigated by water from the Darling River in New South Wales. This was to act as a cross check of area estimates provided by ground personnel of the New South Wales Water Resources Commission and the New South Wales Department of Agriculture. More details of the study and the presentation of some alternative classification techniques for this problem will be found in Moreton and Richards (1984), from which the following sections are adapted.

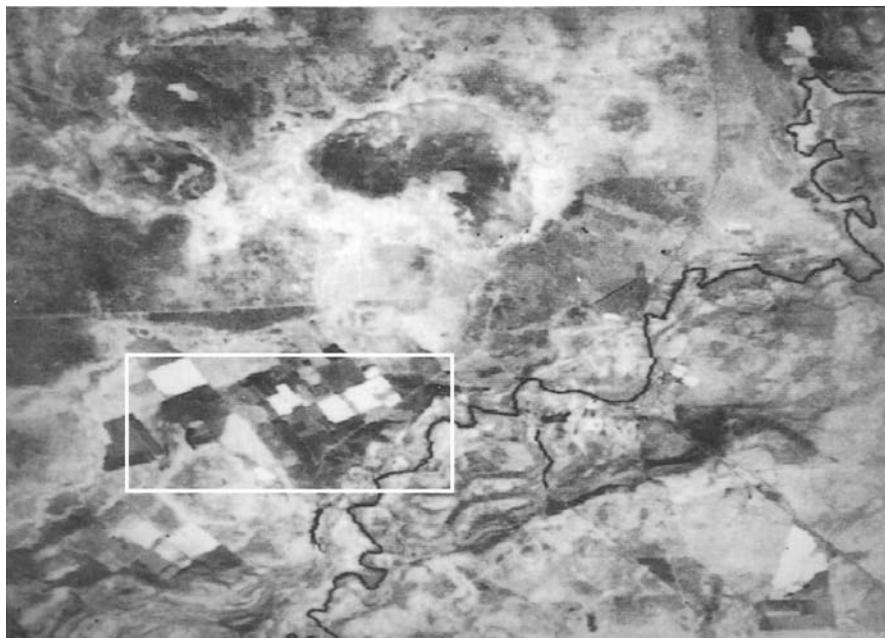
### **11.6.1 Background**

Much of the western region of the state of New South Wales in Australia experiences arid to semi-arid climatic conditions with low average annual rainfalls accompanied by substantial evapotranspiration. Consequently, a viable crop industry depends to a large extent upon irrigation from major river systems. Cotton growing in the vicinity of the township of Bourke is a particular example. With an average annual rainfall of 360 mm, cotton growing succeeds by making use of irrigation from the nearby Darling River. This river also provides water for the city of Broken Hill further downstream and forms part of a major complex river system ultimately that provides water for the city of Adelaide, the capital of the state of South Australia. The Darling River itself receives major inflows from seasonal rains in Queensland, and in dry years can run at very low levels or stop flowing altogether, leading to increased salination of the water supplies of the cities downstream. Consequently, additional demands on the river made by irrigation must be carefully controlled. In New South Wales such control is exercised by the issue of irrigation licenses to farmers. It is then necessary to monitor their usage of water to ensure licenses are not infringed. This, of course, is the situation in many parts of the world where extensive irrigation systems are in use.

The water demand by a particular crop is very closely related to crop area, because most water taken up by a plant is used in transpiration (Keene and Conley, 1980). As a result, it is sufficient to monitor crop area under irrigation as an indication of water used. In this example, classification is used to provide crop area estimates.

### **11.6.2 The Study Region**

A band 7 Landsat Multispectral Scanner image of the region considered in the study, consisting of 927 lines of 1102 pixels, is shown in Fig. 11.1. This is a portion of scene number 30704–23201 acquired in February 1980 (Path 99, Row 81). Irrigated cotton fields are clearly evident as bright fields in the central left and bottom right regions, as is a further crop in the top right. The township of Bourke is just south of the Darling River, just right of the center of the image. The white border encloses a subset of the data, shown enlarged in Fig. 11.2. This smaller region was used for signature generation.



**Fig. 11.1.** Band 7 Landsat MSS image of the region of the investigation, showing irrigated fields (white). The area enclosed by the white border was used for signature generation. Reproduced from *Photogrammetric Engineering & Remote Sensing*. Vol. 50, June 1984

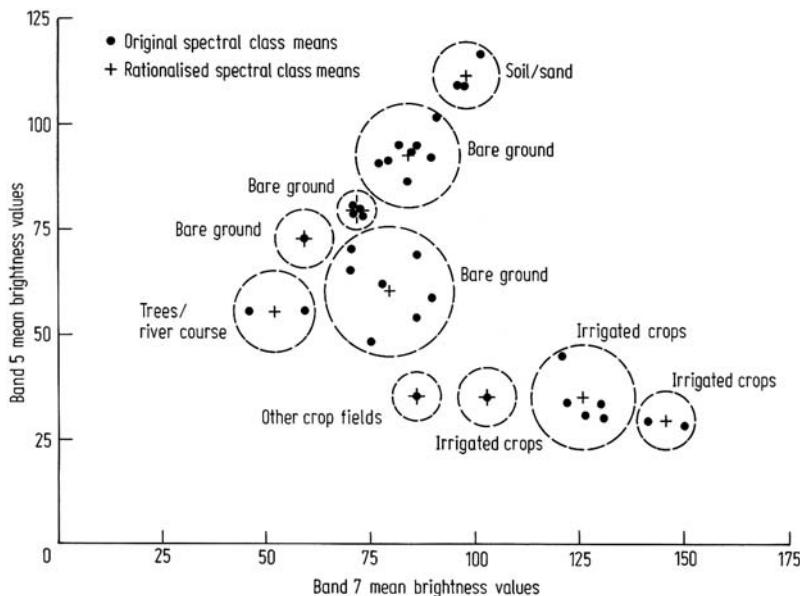
### 11.6.3 Clustering

Figure 11.2 shows the location of four regions selected for clustering using the single-pass algorithm. A fifth clustering region was chosen which partially included the triangular field in the bottom right region of Fig. 11.1. These regions consist of up to 500 pixels each and were selected so that a number of the irrigated cotton fields were included, along with a choice of most of the other major ground covers thought to be present. These include bare ground, lightly wooded regions, such as trees along the Darling River, apparently non-irrigated (and/or fallow) crop land, and a light coloured sand or soil.

Each of the regions shown in Fig. 11.2 was clustered separately. With the parameters entered into the clustering algorithm, each region generated between five and 11 spectral classes. The centres of the complete set of 34 spectral classes were then located on a bispectral plot. Sometimes such a plot could be the average of the visible components of the cluster means (Landsat bands 4 and 5) versus the average of the infrared components (bands 6 and 7). In this exercise, however, owing to the well-discriminated nature of the data, a band 5 versus band 7 bispectral plot was used; moreover, the subsequent classification also made use only of bands 5 and 7. This reduced the cost of the classification phase; however, the results obtained suggest



**Fig. 11.2.** Line printer map (band 7) of the region shown enclosed in a white border in Fig. 11.1. Cluster regions are indicated by the black borders. Reproduced from Photogrammetric Engineering & Remote Sensing, Vol. 50, June 1984



**Fig. 11.3.** Bispectral plot (band 5 class means versus band 7 class means) showing the original 34 cluster centers (spectral classes) generated. Also shown are the class rationalisations adopted. Original spectral classes within the *dotted circles* were combined to form a single class with mean positions indicated. The labels were determined from reference data and spectral response characteristics. *Reproduced from Photogrammetric Engineering & Remote Sensing, Vol. 50, June 1984*

that accuracy was not prejudiced. The band 5 versus band 7 bispectral plot showing the clustering results is illustrated in Fig. 11.3.

At this stage, it was necessary to rationalize the number of spectral classes and to associate spectral classes with ground cover types (so-called information classes). While a sufficient number of spectral classes must be retained to ensure classification accuracy, it is important not to have too many, because the number of class comparisons, and thus the cost of a classification, is directly related to this number. Because the classifier to be employed was known to be of the minimum distance variety, which implements linear decision surfaces between classes, spectral classes were merged into approximately circular groups (provided they were from the same broad cover type) as shown in Fig. 11.3. In this manner, the number of classes was reduced to ten. Labels were attached to each of those (as indicated in Fig. 11.3) by comparing cluster maps to black-and-white and color aerial photography, and to band 7 imagery. The relative band 5 and band 7 brightness values were also employed for class recognition; fields under irrigation were evident by their low band 5 values (30 on a scale of 255, indicating high chlorophyll absorption) accompanied by high band 7 reflectance (100 to 150, indicating healthy, well-watered vegetation).

### 11.6.4

#### Signature Generation

Signatures for the rationalized spectral classes were generated by averaging the means of the constituent original set of spectral classes. This was done manually, and is an acceptable procedure for the classifier used. Minimum distance classification makes use only of class means in assigning pixels and does not take any account of class covariance data. On the contrary, maximum likelihood classification incorporates both class covariance matrices and mean vectors as signatures, and merging of constituent spectral class signatures to obtain those for rationalized classes cannot readily be done by hand. Rather, a routine that combines class statistics is required.

The rationalized class means are indicated in Fig. 11.3.

### 11.6.5

#### Classification and Results

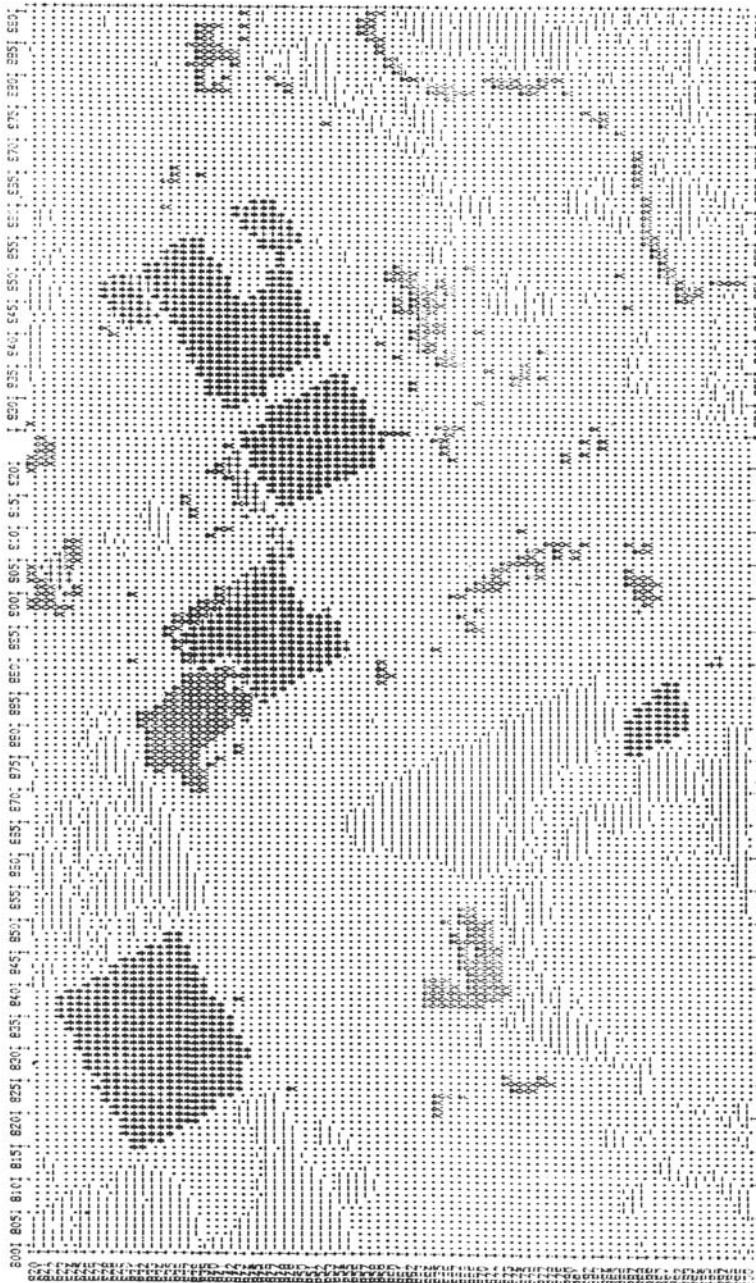
With spectral class signatures determined as above, Fig. 11.1 was checked for crop fields that indicated use of irrigation. A classification map of the Fig. 11.2 (6,957 ha) region is shown in Fig. 11.4. Fields under irrigation are clearly discernible by their shape, as well as by their classification. By retaining several other ground-cover types as separate information classes (rather than giving them all a common symbol representing “non-irrigated”), other geometric features of interest are evident. For example, the Darling River is easily seen, as are some neighbouring fields that are not irrigated. This was useful for checking the results of the classification against maps and other reference data.

The results of the classification agreed remarkably well with ground-based data gathered by field officers of the New South Wales Water Resources Commission and the New South Wales Department of Agriculture. In particular, for a region of 169651 pixels (75,000 ha) within Fig. 11.1, a measure of 803 ha given by the classifier as being under irrigation agreed to better than 1% with that given by ground data. This is well within any experimental error that could be associated with the classification and with the uncertainty regarding pixel size (in hectares), and is consistent with accuracies reported by some other investigators (Tinney et al., 1974).

### 11.6.6

#### Concluding Remarks

In general, the combined clustering/supervised classification strategy adopted works well as a means for identifying a reliable set of spectral classes upon which a classification can be based. The clustering phase, along with a construction such as a bispectral plot, is a convenient and lucid means by which to determine the structure of image data in multispectral space; this would especially apply for exercises that are as readily handled as those described here. The rationalized spectral classes used in this case correspond not so much to unimodal Gaussian classes normally associated with maximum likelihood classification, but rather are a set that match the



**Fig. 11.4.** Classification map of the region of Fig. 11.2 generated using the ORSER software package. Class symbols used are:  
\* irrigated crops; + other crop fields; X trees/river course; - soil/sand; bare ground. *Reproduced from Photogrammetric Engineering & Remote Sensing, Vol. 50, June 1984*

characteristics of the minimum distance classifier employed. This is an important general principle: the analyst should know the properties and characteristics of the classifier being used and, from a knowledge of the structure of the image, choose spectral class descriptions that match the classifier.

## 11.7

### Case Study 2: Multitemporal Monitoring of Bush Fires

This case study demonstrates three digital image processing operations: image-to-image registration, principal components transformation and unsupervised classification. It entails the use of two Landsat multispectral scanner image segments of a region in the northern suburbs of the city of Sydney, New South Wales. The region is subject to damage by bush fires, and the images show fire events and revegetation in the region over a period of twelve months. Full details of the study can be found in Richards (1984) and Richards and Milne (1983).

#### 11.7.1

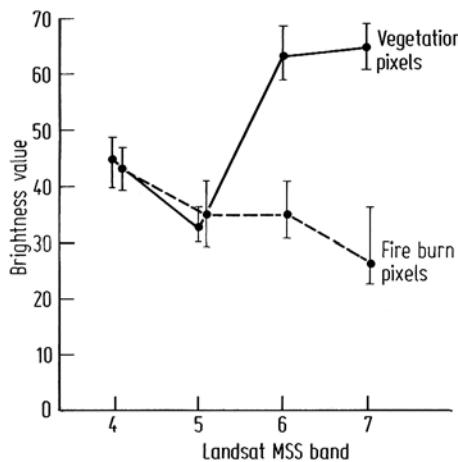
##### Background

The principal components transformation developed in Chap. 6 is a redundancy reduction technique that generates a new set of variables with which to describe multispectral remote sensing data. These new variables, or principal components, are such that the first contains most of the variance in the data, the second contains the next major portion of variance and so on. Moreover, in these principal component axes the data is uncorrelated. Owing to this it has been used as a data transform to enhance regions of localised change in multitemporal multispectral image data (Byrne and Crapper 1979; Byrne et al., 1980; Ingebritsen and Lyon 1985; Fung and Le Drew 1987). This is a direct result of the high correlation that exist between image data for regions that do not change significantly and the relatively low correlation associated with regions that change substantially. Provided the major portion of the variance in a multitemporal image data set is associated with constant cover types, regions of localised change will be enhanced in the higher components of the set of images generated by a principal components transformation of the multitemporal, multispectral data. Since bushfire events will often be localised in image data of the scale of Landsat multispectral scanner imagery, the principal components transformation should therefore be of value as a preclassification enhancement (and, as it transpires, as a feature reduction tool).

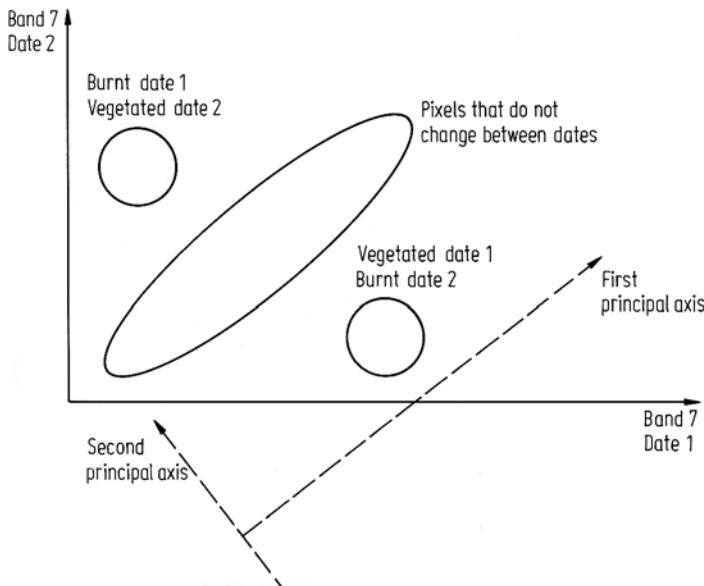
#### 11.7.2

##### Simple Illustration of the Technique

Figure 11.5 shows the spectral reflectance data of healthy vegetation and vegetation damaged by fire, typical of that in the image data to be used below. As expected,



**Fig. 11.5.** Typical spectral reflectance data of healthy vegetation pixels and fire damaged vegetation pixels. These have been derived from the actual image data used below. *Reproduced from Remote Sensing of Environment, Vol. 16, 1984*



**Fig. 11.6.** Hypothetical illustration of a 2 dimensional 2 date Landsat MSS band 7 space, showing the dispersion of pixels associated with constant cover types and those that change between dates

the major effect is in the infrared region, corresponding to band 7 of the Landsat (1–3) MSS. To illustrate the value of principal components in highlighting changes associated with fire events suppose we consider just band 7 data from two dates. One date is prior to a fire and the other afterwards. We can construct a two date scatter diagram as shown in Fig. 11.6. Pixels that correspond to cover types that remain essentially constant between dates cluster about an elongated area as shown,

representing water, vegetation and soils. Cover types that change between dates show as major departures from that general trend. For example pixels that were vegetated in the first date and burnt in the second lead to the off-diagonal cluster shown. Similarly pixels that appeared burnt in the first date and revegetated in the second appear as an off-diagonal cluster in the opposite direction.

Principal components analysis will lead to the axes shown in Fig. 11.6. As seen the band 7 variations associated with the localised changes project into both component axes. However the effect is masked in the first component by the large range of brightnesses associated with the near-constant cover types. By comparison the change effect in the second component will dominate since the constant cover types will map to a small range of brightness in the second principal component.

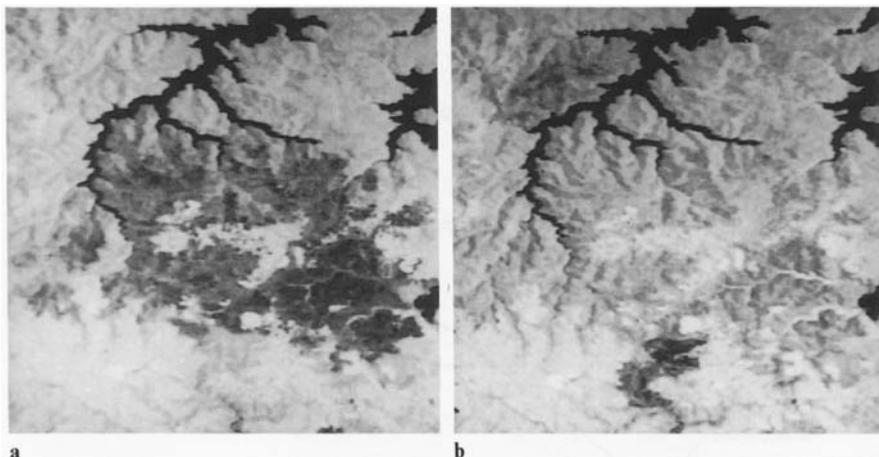
The same general situation occurs when all available bands of image data are used. However several of the higher order components will reflect local change information.

### **11.7.3 The Study Area**

In December 1979 the State of New South Wales in Australia experienced a number of major bushfires. While the majority of these were in mountain ranges to the west and northwest of the capital, Sydney, one particularly threatening fire occurred in Sydney's northern suburbs. Figure 11.7a shows a portion of a Landsat MSS image in this vicinity acquired on 29 December 1979. The area of bushland damaged by fire appears dark. The same region almost one year later (14 December 1980) is shown in Fig. 11.7b. The 1979 fire scar is diminished owing to partial vegetation recovery. However, two new fire burns are evident, as indicated, resulting from fires in the intervening period. The pair of images together therefore allow examination of vegetation to fire burn change and fire burn to revegetation.

### **11.7.4 Registration**

Areas about two to three times larger than those shown in Fig. 11.7 were registered utilizing approximately 20 control points spaced near the scene circumference with a few scattered over the scene centre. The actual positions of some control points used are shown in Fig. 2.17a by comparison to the study area. Cubic polynomial mapping of the 1980 image to the 1979 image was performed. Resampling was based upon cubic convolution interpolation since the primary intention of the project was to examine principal components images by photointerpretation. The resulting average standard errors for the prediction of control points in the master image from those in the slave were less than 1/4 pixel spacing in both row and column.

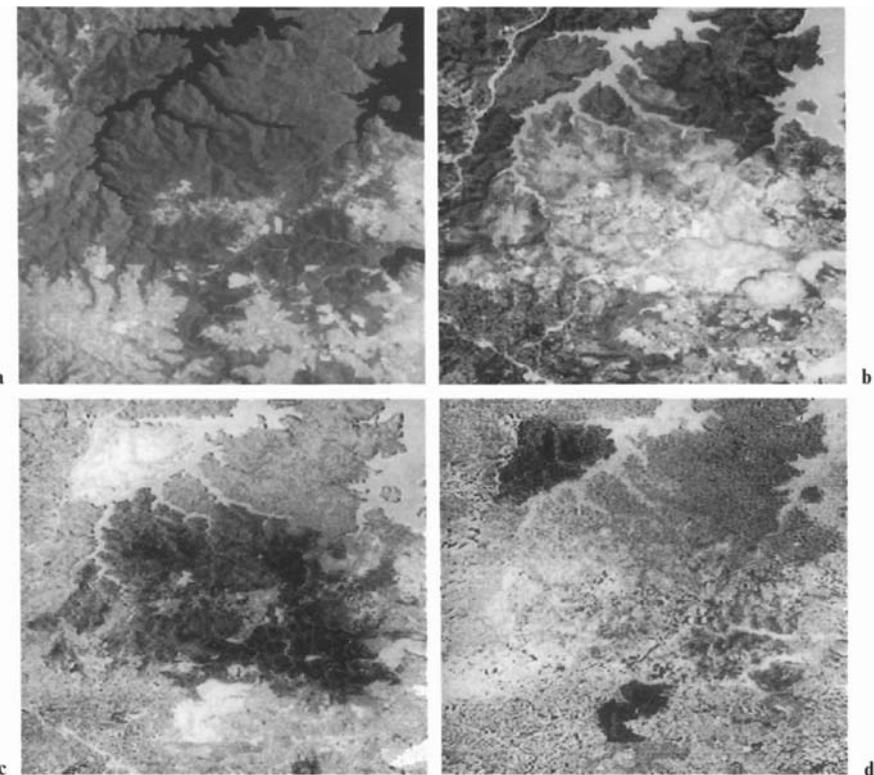


**Fig. 11.7.** **a** Portion of the Landsat MSS band 7 data acquired over Sydney on 29 December 1979. The dark region in the centre is an area burnt out by a fire several days earlier. **b** The Landsat MSS band 7 of the region in **a** but acquired almost 1 year later, on 14 December 1980. The fire burn is revegetating as evidenced by the developing light regions. Two new fire burns have occurred. These show as dark regions on the north-western and southern regions of the image segment

### 11.7.5 Principal Components Transformation

The registered subscenes were added to form an 8-dimensional multitemporal image data set (in the order 1979 band 4, 1979 band 5 . . . , 1980 band 4, 1980 band 5 . . . ), from which the set of principal components was generated. Automatic polarization (inversion of brightnesses) and scaling options were chosen in the transformation process as these gave component images with better visual dynamic range and it was felt that they would not prejudice subsequent interpretation at the level of discrimination envisaged (into major cover types and change classes but not into fine subdivision of vegetation species, etc.).

The first four of the eight principal component images are shown in Fig. 11.8. The remainder do not display any features of significance to the study. The first component is tantamount to a total brightness image, whereas the later components highlight changes. It is the second, third and fourth components that are most striking in relation to the fire features of interest. Pixels that have essentially the same cover type in both dates e.g., vegetation and vegetation, fire burn and fire burn, show as midgrey in the second, third and fourth components. Those that have changed, either as vegetation to fire burn or as fire burn to vegetation show as darker or brighter than midgrey, depending upon the component. These effects are easily verified by substituting typical spectral reflectance characteristics into the equations that generate the components. Each component is a linear combination of the original eight bands of data, where the weighting coefficients are the components of the corre-



**Fig. 11.8.** The first four principal components of the 8-dimensional data set formed by concatenating the four Landsat MSS bands of the region of interest from each date. Components are numbered as **a** PC 1; **b** PC 2; **c** PC 3; **d** PC 4. Components 3 and 4 particularly highlight the fire-related events. *Reproduced from Remote Sensing of Environment, Vol. 16, 1984*

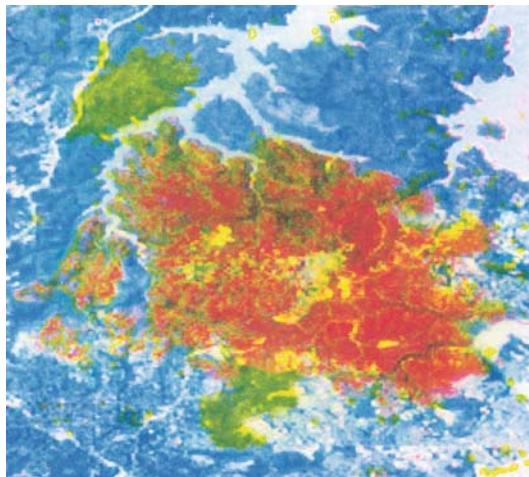
sponding eigenvector of the  $8 \times 8$  covariance matrix. These eigenvectors along with their associated eigenvalues (which are the variances of the components) are shown in Table 11.4. In interpreting the fourth component it is necessary to account for a polarization inversion introduced in generating the set of principal components.

The second principal component image expresses the 1979 fire burn as lighter than the average image tone, while the third principal component highlights the two fire burns. The 1979 burn region shows as darker than average whereas that for 1980 shows as slightly lighter than average. In the fourth component the 1980 fire burn shows as darker than average with the 1979 scar not evident. What can be seen, however, is revegetation in 1980 from the 1979 fire. This shows as lighter regions. A particular example is revegetation in two stream beds on the right-hand side of the image a little over halfway down.

A colour-composite image formed by displaying the second principal component as red, the third as green, and the fourth as blue is shown in Fig. 11.9. This shows the area that was vegetated in 1979 but burned in 1980 as lime green; the regions

**Table 11.4.** Eigenvalues (variances) and eigenvectors of the 8-dimensional, original image data covariance matrix. The eigenvectors are the component weighting coefficients

Component	Eigenvalue	Eigenvector								
1	1884	0.14	0.21	0.38	0.38	0.15	0.30	0.53	0.50	
2	236	0.24	0.32	-0.21	-0.45	0.36	0.63	0.06	-0.25	
3	119	0.24	0.21	0.49	0.46	0.07	0.08	-0.40	-0.53	
4	19	-0.51	-0.58	-0.03	0.27	0.13	0.55	-0.04	-0.12	
5	6	0.37	-0.50	0.07	-0.04	0.38	-0.30	0.49	-0.37	
6	5	0.44	-0.14	-0.54	0.41	0.31	0.00	-0.37	0.32	
7	4	-0.17	0.35	-0.52	0.45	-0.19	-0.05	0.42	-0.39	
8	3	0.50	-0.29	-0.04	-0.02	-0.74	0.34	0.08	-0.04	



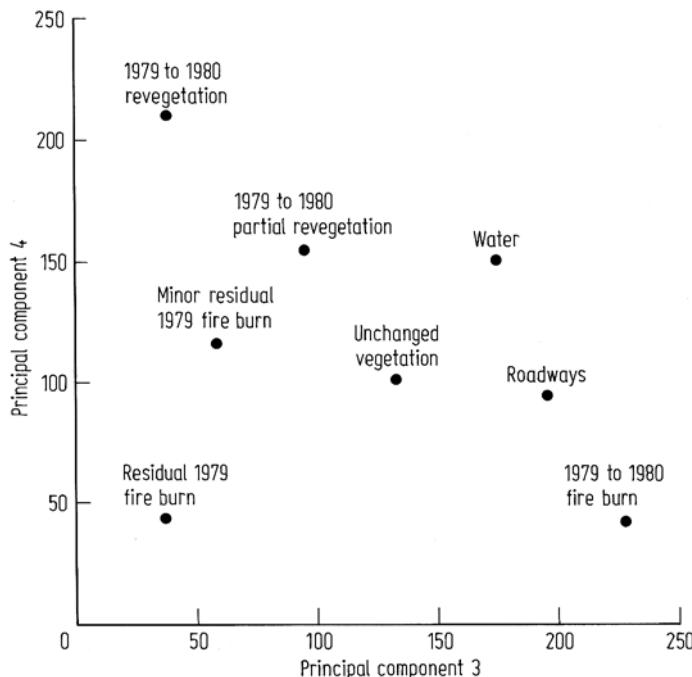
**Fig. 11.9.** Colour composite multitemporal image formed by displaying the second principal component as red, the third principal component as green and the fourth principal component as blue

from the 1979 burn that remain without vegetation or have only a light vegetation cover in 1980 show as bright red; revegetated regions in 1980 from the 1979 fire display as bright blue/purple whereas the vegetated, urban, and water backgrounds that remained essentially unchanged between dates show as dark green/grey.

### 11.7.6 Classification of Principal Components Imagery

Because of the change enhancement offered in the principal components it should be possible to produce a change class thematic map by classification.

An initial unsupervised classification of the first four principal components produced substantial confusion between water/land and fire burn/vegetation. Owing to the nature of the first component, this is to be expected. A second test using com-

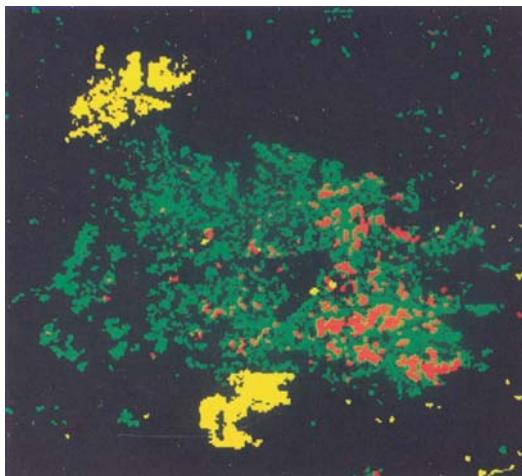


**Fig. 11.10.** Bispectral plot used for developing spectral classes in the classification of the third and fourth principal components into major fire-related themes. *Reproduced from Remote Sensing of Environment, Vol. 16, 1984*

ponents 2, 3, and 4 was acceptable, although some of the richly revegetated regions were unclassified. Consequently, it was decided to use just components 3 and 4 in the classification since a visual inspection indicates that they contain most of the class/change class information required.

The six major cover types were roadways, water, 1979 to 1980 fire burn, 1979 to 1980 revegetation, unchanged vegetation, and residual 1979 fire burn. Unchanged urban regions were not considered since resolution of these from other unchanged classes such as vegetation and water was not required. Maximum likelihood signatures for the six selected classes were generated. This left a significant proportion of what could be called “partial revegetation (1979 to 1980)” and “minor residual 1979 fire burn” unclassified. This situation was rectified by adding these two further classes as shown in the bispectral plot of Fig. 11.10.

The classification map of Fig. 11.11 was obtained using the eight subclasses, along with a likelihood threshold so chosen to avoid classification of regions for which signatures were not developed (such as urban). In the map only three major change classes are displayed, these being minor and major vegetation regeneration from 1979 to 1980 (the decision was made by inspection of the original standard colour composite images), the 1980 fire burn, and the residual bare effect from the 1979 fire. The latter is not strictly a change class for the pair of images considered



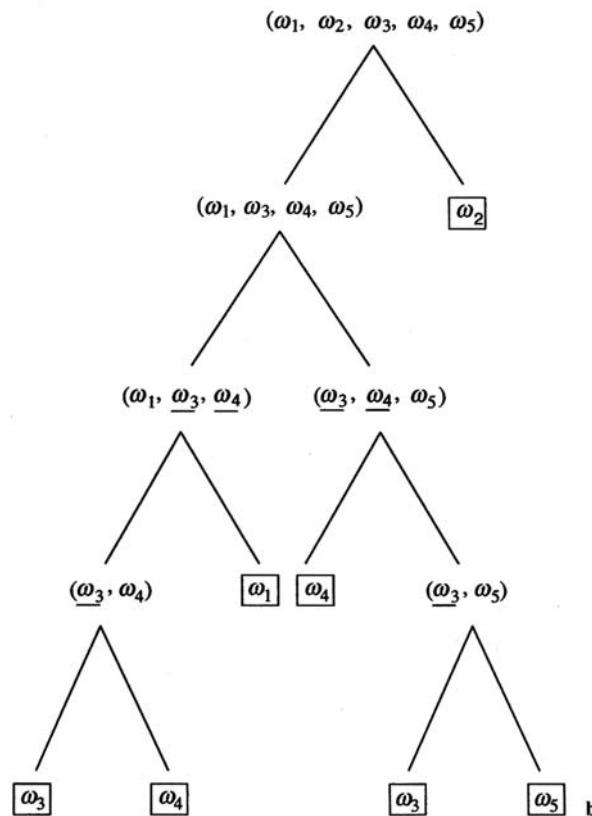
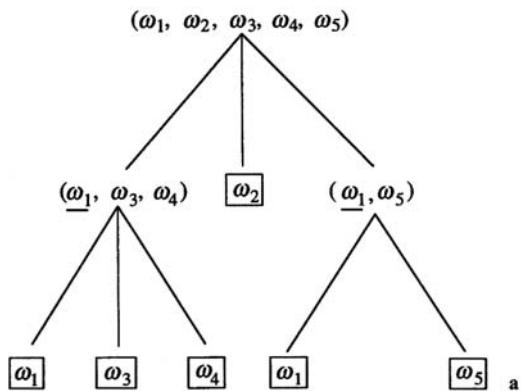
**Fig. 11.11.** Thematic map generated by classification of the third and fourth principal components. Only three classes are shown. These are vegetation regeneration (green), the 1980 fire burn (yellow) and the residual bare effect from the 1979 fire (red)

but, nevertheless, was generated easily and is a significant class in the context of the study of fire damage and vegetation recovery. Notable in this classification is that there appears to be no confusion between burn and revegetating pixels, and water edge regions. The reason for this is that the water edge pixels are approximately constant between dates (to the extent that tides are constant) and thus are correlated. They will map therefore to the midgrey constant background region of the higher order principal components, whereas the fire burn pixels (with which they can be confused) are vegetated in one date and burned in another and thus map to a quite different range of brightness in transformed imagery. Experience with single scene classification often shows water edge and fire burn confusion.

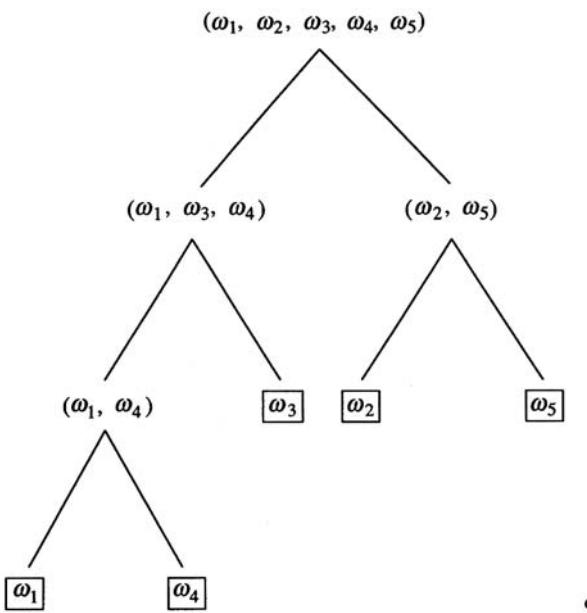
## 11.8 Hierarchical Classification

### 11.8.1 The Decision Tree Classifier

The classifiers treated in above have all been single stage in that only one decision is made about a pixel, as a result of which it is labelled as belonging to one of the available classes or is left unclassified. Multistage classification techniques are also possible in which a series of decisions is taken in order to determine the correct label for a pixel. Examples of such an approach are shown in Figs. 8.17 and 8.18. The more common multistage classifiers are called decision trees, examples of which are shown in Fig. 11.12. They consist of a number of connected classifiers (or decision



**Fig. 11.12.** **a** A general decision tree. **b** A binary decision tree with overlapping classes. **c** A binary tree without overlapping classes – underlines indicate class overlaps



**Fig. 11.12c.**

nodes) none of which is expected to perform the complete segmentation of the image data set. Instead, each component classifier only performs part of the task, as noted in the figure. Perhaps the simplest type is the binary tree in which each component classifier, or node, is expected to perform a segmentation of the data into only one of two possible classes, or groups of classes.

The advantages of using a multistage or tree approach to classification include that different data sources, different sets of features, and even different algorithms can be used at each decision stage. Minimising the number of features to use in a decision is significant for reducing processing time and for improving the accuracy of small class training.

## 11.8.2 Decision Tree Design

Frequently, decision tree strategies can be designed manually, particularly when they are required to perform quite specific labelling tasks (Swain and Hauska, 1977). However, as with single stage classifier and neural network training it would be of value to have automated design procedures available.

There are three tasks in the design of a decision tree: finding the optimal structure for the tree, choosing the optimal subset of features at each node, and selecting the decision rule to use at each node. An optimal or suboptimal tree structure may aim for minimum error rate, a minimum number of nodes, or a minimum path length in deciding how to split classes at each node of the tree; consideration must be given also

to means for controlling overlapping classes and for control of how many branches and layers to use.

Since the number of possible tree structures, even for a moderately small number of classes, is astronomical, it is very difficult to design an optimal classifier (Mui and Fu, 1980). Classification accuracy and efficiency, however, rely heavily on the tree chosen. Therefore, various heuristic methods for decision tree design have been developed, details of which can be found in Safavian and Landgrebe (1991).

To make the design task easier, binary decision trees are often adopted. Discrimination ability is not necessarily weakened by choosing a binary approach, since a general decision tree can be uniquely transformed into an equivalent binary tree (Rounds, 1980).

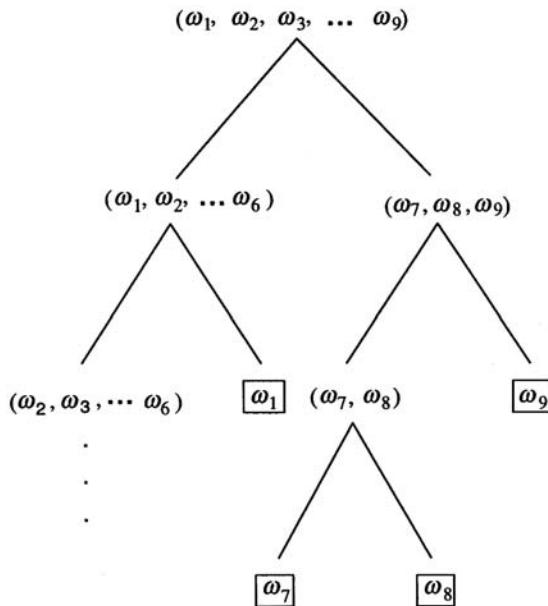
One method for binary decision tree design is a “bottom up” approach similar to the agglomerative hierarchical clustering algorithm discussed in Sect. 9.7. If we replace the pixel data by class mean vectors, the bottom up method can be implemented by that process. Initially, the pairwise class separations are computed using a suitable distance metric, such as Euclidean distance. The two most similar classes are merged and a new mean is estimated for this combined data. This is continued until all the classes lie in a single, large class. The history of mergings provides the inverse order of classes split in the decision tree.

Figure 11.13 shows the decision tree corresponding to the data given in Fig. 9.7. (The 9 pixels are treated as 9 class mean vectors.) As seen, the two most separable groups of classes are processed first, and the most subtle class pair will be discriminated at the bottom of the tree. By so doing, the cumulative error will also be minimised.

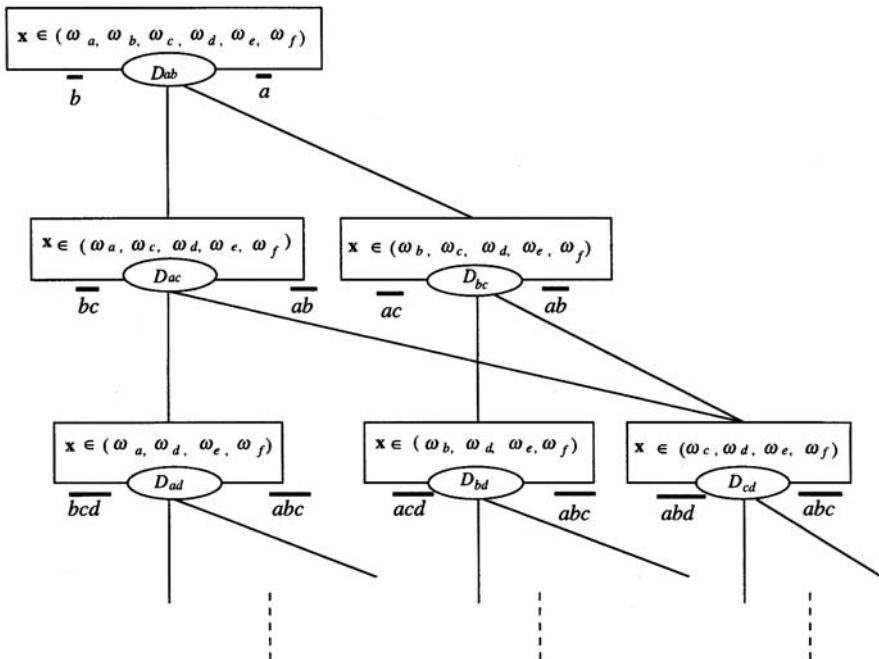
This method assumes that the same set of features and the same classification algorithm are used at each decision node. A more general design philosophy is difficult to devise. However, analyst knowledge often helps in structuring a tree. For example, it is logical to separate data into water and croplands, and then croplands further into wheat, corn, etc. A user might also be able to use algorithm knowledge such as that minimum distance classification is preferred when small classes need to be identified. Moreover, some GIS data, e.g. elevation, can be segmented by a one dimensional parallelepiped algorithm.

### 11.8.3 Progressive Two-Class Decision Classifier

Figure 11.14 shows a progressive two-class decision classifier (Jia and Richards, 1998). Suppose there are six classes, represented by  $\omega_a, \omega_b, \omega_c, \omega_d, \omega_e$  and  $\omega_f$ . The scheme focuses on one class pair at a time (at a node). The function of the first layer is to check the potential membership of an unknown pixel vector  $x$  to class  $\omega_a$  and  $\omega_b$  and the vector is classified temporarily as either class  $\omega_a$  or class  $\omega_b$  using the decision rule,  $D_{ab}$ . Class  $\omega_b$  will be rejected for further consideration for those vectors labelled into the  $\omega_a$  category, and class  $\omega_a$  is rejected for further consideration for those vectors labelled into the  $\omega_b$  category. At the second layer, there are two nodes, and two new class pairs ( $\omega_a$  and  $\omega_c$  for the left side node and  $\omega_b$  and  $\omega_c$  for



**Fig. 11.13.** The binary decision tree for the data shown in Fig. 9.7



**Fig. 11.14.** A schematic chart for progressive two-class decision classifier

the right side node) are considered, respectively. This process continues until a pure class labelling has been reached at the last layer, which is the final assignment.

Since the class pair considered at each node is clear, one can concentrate on making decisions on which algorithm and which subset features to use for the particular class pair. An optimal environment for discriminating individual class pairs may result and thus maximum separation between them achieved.

As an example of how the progressive two-class decision tree can lead to good results an AVIRIS image of mixed agriculture and forestry in Northwestern Indiana, USA recorded in June 1992 was chosen for classification. Water absorption bands, bands 104 to 108 and 150 to 162, were removed, leaving 202 of the original 220 bands.

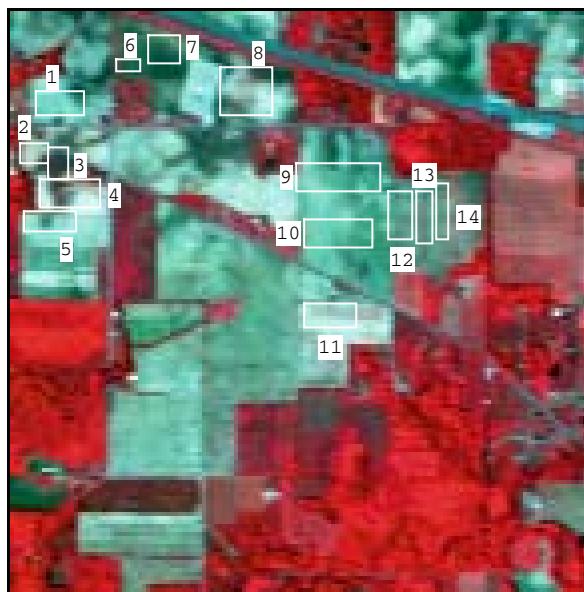
The number of bands is too high relative to the available training samples to generate reliable class statistics. Therefore, a principal components transformation was applied to the data and the first 40 transformed features, containing 99.93% of total variance, were retained for the exercise.

Four difficult-to-separate classes were chosen. Since the data was collected in the early part of the growing season, soybean and corn canopies presented only about a 5% ground cover (Landgrebe, 1995). These two classes were each divided into two subcategories depending upon the tillage practice used on a given field: no-till and clean-till. The no-till fields would have a substantial amount of residue from the crop of the previous year and the clean-till fields would have a background of soil only. The separation of these four classes, two species in each of two conditions, represents a challenging classification problem. The image, and details of these four classes, are given in Fig. 11.15.

Gaussian maximum likelihood classification was run on the selected pixels. Class statistics were estimated using the training pixels. The best features were found for each class pair separation as shown in Table 11.5. It can be seen that the most suitable (and number of) features to use for one class pair are often not the same as for other class pairs. When all classes are to be separated in a single step, 9 features gave the best result of 64.4% on the testing data. In contrast, using a progressive two-class decision tree an accuracy of 72.5% was obtained.

**Table 11.5.** The best features found for each class pair and for all classes considered together

Class Pairs	No. of Features Selected	Best Features Selected
Classes 1 & 2	3	1, 2 & 14
Classes 1 & 3	3	2, 3 & 6
Classes 1 & 4	2	14 & 15
Classes 2 & 3	3	1, 2 & 3
Classes 2 & 4	1	19
Classes 3 & 4	3	2, 14 & 15
All classes together	9	1, 2, 3, 6, 10, 12, 14, 15 & 19



Class Number	Class Name	Training Fields	Number of Training Pixels	Testing Fields	Number of Testing Pixels
1	Corn no-till	11, 13	130	1, 12	144
2	Corn clear-till	4	105	2, 3	75
3	Soybean no-till	6, 10	137	9, 14	189
4	Soybean clear-till	5, 7	121	8	156

**Fig. 11.15.** The image, training and testing pixels used in the classification exercise

#### 11.8.4 Error Accumulation in a Decision Tree

A single layer classification (for example, maximum likelihood on the complete set of features to resolve the data into all the information classes in one step) can be represented by a binary decision tree. When a fixed set of features and decision rule are used at every node, the binary tree can in fact be shown to be identical to single layer classification (Mui and Fu, 1980). When optimal or suboptimal features and the most appropriate decision boundaries are used at each stage, classification performance might be improved as demonstrated by Swain and Hauska (1977), Iikura and Yasuoka (1991), Lee and Richards (1985) and Kim and Landgrebe (1991).

However, improved performance is not always achieved. Unfortunately with a decision tree there will be an accumulation of error, thereby requiring very good decisions at each node if acceptable classification errors are to be maintained at the terminal nodes. This can be seen in the following simple analysis for a binary tree.

Suppose the probability of error for both outcomes at every node of a binary tree is  $p$ . Although this is simplistic it serves to illustrate the problem. For a classification requiring only a single decision node then the error will be  $p$ .

However if two decision nodes are crossed in determining the labelling for a pixel then the (accumulated) error will be

$$p_E = p + (1 - p)p = 2p - p^2 \quad (\text{two decision nodes})$$

The first term in this expression represents erroneous decisions from the previous node while the second is the probability of correct classification from the previous node ( $1 - p$ ) multiplied by the probability of making an error on those correctly classified pixels at the second node. For a separation requiring three nodes of decision the accumulated error, proceeding in the same manner, will be

$$p_E = 2p - p^2 + (1 - 2p + p^2)p = 3p - 3p^2 + p^3 \quad (\text{three decision nodes})$$

Generalising, it can be shown that the error accumulated after  $N$  nodes of decision will be

$$p_E = 1 - (1 - p)^N$$

As an example, if  $p = 10\%$ ,  $N = 5$ , the final error will be 59%. Thus, owing to the effect of error accumulation, it is critical to ensure there is very high classification accuracy at individual nodes in order to maintain a satisfactory accuracy at the end of the tree. Hopefully this effect will be obviated to an extent by the fact that the algorithm and features used at each node may be optimal or near optimal for the separation to be performed at the node.

A further reason as to why binary tree classifiers do not necessarily improve the correct recognition rate is that when some classes are merged into a group at a node, the decision boundaries become less specific in the discrimination between mixtures of classes (Landeweerd et al., 1983). Kim and Landgrebe (1991) point out that, if there were no Hughes phenomenon, the single-layer maximum likelihood classifier would have better performance than any decision tree classifier.

## 11.9

### A Note on Hyperspectral Data Classification

The focus of the classification methodologies treated in this chapter has been on data sets where the number of spectral bands is not high. When hyperspectral data needs analysis major difficulties can arise with standard classification procedures if sufficient training samples are not available to estimate class signatures reliably. To an extent, the problem can be eased if decision trees are used, so that not all features are used at each decision node. Generally, however, when the number of bands or channels is large, either care needs to be taken when using standard parametric procedures (see Chap. 13) or else quite different analytical approaches need to be adopted, including the use of expert knowledge of spectroscopic principles (see Chap. 12).

## References for Chapter 11

A good discussion on choice of sampling strategies and means for determining reliable reference samples for assessing thematic map accuracy will be found in Stehman and Czaplewski (1998). Richards (1996) draws attention to the distinction between the performance of a classifier and the accuracy of the resulting thematic map, and notes conditions under which the two are the same.

Many of the international journals and conferences devoted to remote sensing technology contain case studies dealing with the use of satellite and aircraft acquired digital image data in a variety of applications. Journals that could be consulted include *Remote Sensing of Environment*, *Photogrammetric Engineering and Remote Sensing*, *International Journal of Remote Sensing*, *ISPRS Journal of Photogrammetry and Remote Sensing*, *Geocarto* and the *IEEE Transactions on Geoscience and Remote Sensing*. Relevant conferences include Purdue University's Symposia on Machine Processing of Remotely Sensed Data from the 1960's to the 1980's that place emphasis on digital processing techniques, the International Symposia on Remote Sensing of Environment run by the Environmental Research Institute of Michigan, and the IEEE International Geoscience and Remote Sensing Symposia.

The decision trees treated in this chapter can be viewed as specific examples of the general class of layered classifiers (Nilsson, 1965, 1990). Another is a piecewise linear classifier, such as that proposed by Lee and Richards (1985), which is based upon a decision tree of the threshold logic units discussed in Sect. 8.10.1.

- F.C. Billingsley, 1982: Modeling Misregistration and Related Effects on Multispectral Classification. *Photogrammetric Engineering and Remote Sensing*, 48, 421–430.
- G.R. Byrne and P.F. Crapper, 1979: An Example of the Detection of Changes between Successive Landsat Images by Numerical Methods in an Urban Area. Proc. 1st Australasian Conference on Remote Sensing (Landsat '79), Sydney.
- G.R. Byrne, P.F. Crapper and K.K. Mayo, 1980: Monitoring Land Cover Changes by Principal Components Analysis of Multitemporal Landsat Data. *Remote Sensing of Environment*, 10, 175–184.
- R. Congalton and K. Green, 1999: Assessing the Accuracy of Remote Sensing Data: Principles and Practices. Roca Baton, Lewis.
- M.D. Fleming, J.S. Berkebile and R.M. Hoffer, 1975: Computer Aided Analysis of Landsat-1 MSS Data: A Comparison of Three Approaches including a Modified Clustering Approach. Information Note 072475, Laboratory for Applications of Remote Sensing, West Lafayette, Indiana.
- J.E. Freund, 1992: Mathematical Statistics 5e, Englewood Cliffs, N.J., Prentice-Hall.
- B.C. Forster and J.C. Trinder, 1984: An Examination of the Effects of Resampling on Classification Accuracy. Proc 3rd Australasian Conf. on Remote Sensing (Landsat '84), Gold Coast, Queensland, 106–115.
- T. Fung and E. LeDrew, 1987: Application of Principal Components Analysis to Change Detection. *Photogrammetric Engineering and Remote Sensing*, 53, 1649–1658.
- R.M. Hord and W. Brooner, 1976: Land-Use Map Accuracy Criteria. *Photogrammetric Engineering and Remote Sensing*, 42, 671–677.
- W.D. Hudson and C.W. Ramm, 1987: Correct Formulation of the Kappa Coefficient of Agreement. *Photogrammetric Engineering and Remote Sensing*, 53, 421–422.
- Y. Iikura and Y. Yasuoka, 1991: Utilisation of a Best Linear Discriminant Function for Designing the Binary Decision Tree. *Int. J. Remote Sensing*, 12, 55–67.

- S.E. Ingebritsen and R.J.P. Lyon, 1985: Principal Components Analysis of Multitemporal Image Pairs. *Int. J. Remote Sensing*, 6, 687–696.
- X. Jia and J.A. Richards, 1998: Progressive Two-class Decision Classifier for Optimization of Class Discriminations. *Remote Sensing of Environment*, 63, 289–297.
- K.M. Keene and C.D. Conley, 1980: Measurement of Irrigated Acreage in Western Kansas from Landsat Images. *Environmental Geology*, 3, 107–116.
- B. Kim and D.A. Landgrebe, 1991: Hierarchical Classifier Design in High-dimensional, Numerous Class Cases. *IEEE Trans. on Geoscience and Remote Sensing*, 29, 518–528.
- G.H. Landeweerd, T. Timmers, E.S. Gelsema, M. Bins and M.R. Halie, 1983: Binary Tree Versus Single Level Tree Classification of White Blood Cells. *Pattern Recognition*, 16, 571–577.
- D.A. Landgrebe, 2003: Signal Theory Methods in Multispectral Remote Sensing, N.J., Wiley.
- T. Lee and J.A. Richards, 1985: A Low Cost Classifier for Multitemporal Applications. *Int. J. Remote Sensing*, 6, 1405–1417.
- G.E. Moreton and J.A. Richards, 1984: Irrigated Crop Inventory by Classification of Satellite Image Data. *Photogrammetric Engineering and Remote Sensing*, 50, 729–737.
- J.K. Mui and K.S. Fu, 1980: Automated Classification of Nucleated Blood Cells Using a Binary Tree Classifier. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-2, 429–443.
- N.J. Nilsson, 1965: Learning Machines. N.Y., McGraw-Hill.
- N.J. Nilsson, 1990: The Mathematical Foundations of Learning Machines. Palo Alto, Morgan Kaufmann.
- J.A. Richards and A.K. Milne, 1983: Mapping Fire Burns and Vegetation Regeneration Using Principal Components Analysis. Proc. 1983 Int. Geoscience and Remote Sensing Symposium. San Francisco.
- J.A. Richards, 1984: Thematic Mapping from Multitemporal Image Data Using the Principal Components Transformation. *Remote Sensing of Environment*, 16, 35–46.
- J.A. Richards, 1996: Classifier Performance and Map Accuracy. *Remote Sensing of Environment*, 57, 161–166.
- G.H. Rosenfield, K. Fitzpatrick-Lins and H.S. Ling, 1982: Sampling for Thematic Map Accuracy Testing. *Photogrammetric Engineering and Remote Sensing*, 48, 131–137.
- E.M. Rounds, 1980: A Combined Nonparametric Approach to Feature Selection and Binary Decision Tree Design. *Pattern Recognition*, 12, 313–317.
- S.R. Safavian and D.A. Landgrebe, 1991: A Survey of Decision Tree Classifier Methodology. *IEEE Trans. on System, Man, and Cybernetics*, 21, 660–674.
- S.V. Stehman and R.L. Czaplewski, 1998: Design and Analysis for Thematic Map Accuracy Assessment, Fundamental Principles. *Remote Sensing of Environment*, 64, 331–344.
- P.H. Swain and H. Hauska, 1977: The Decision Tree Classifier: Design and Potential. *IEEE Trans. on Geoscience Electronics*, GE-15, 142–147.
- L.R. Tinney, J.E. Estes, K.H. Thaman and R.R. Thaman, 1974: Operational Use of Satellite and High Altitude Remote Sensing for the Generation of Input Data for Water Demand Models. Proc. 9th Int. Symp. on Remote Sensing of Environment, Michigan, 739–757.
- J.L. Van Genderen, B.F. Lock and P.A. Vass, 1978: Remote Sensing: Statistical Testing of Thematic Map Accuracy. *Remote Sensing of Environment*, 7, 3–14.
- J. Verdin, 1983: Corrected vs Uncorrected Landsat 4 MSS Data. *Landsat Data Users Notes*, Issue No. 27, Sioux Falls, NOAA, June 4–8.

## Problems

**11.1** (a) What is the difference between an *information class* and a *spectral class*?  
 (b) Four analysts use different quantitative methods for analysing multispectral satellite data. These are summarised below. Comment on the merits and shortcomings of the four approaches and indicate which one you think is most effective.

Analyst 1

1. Chooses training data from homogeneous regions for each cover type.
2. Develops statistics for a maximum likelihood classifier.
3. Classifies image.

Analyst 2

1. Performs a clustering of the whole image and attaches labels to each cluster type afterwards.

Analyst 3

1. Chooses several regions within the image, each of which includes more than one cover type.  
 Clusters each region.
2. Identifies the cluster types.
3. Uses statistics from the clustering process to perform a maximum likelihood classification of the whole image.

Analyst 4

1. Chooses training fields within apparent homogenous regions for each cover type. Clusters those regions to identify spectral classes.
  2. Uses statistics from the clustering process to perform a maximum likelihood classification of the whole image.
- (c) For the method you have identified in (b) as best, discuss how separability measures could be included to advantage.

**11.2** The spectral classes used with the maximum likelihood decision rule in supervised classification are assumed to be representable by single multivariate normal probability distributions. Geometrically, this implies that they will have a hyperellipsoidal distribution in multispectral space. Do you think clustering by the iterative moving means algorithm will generate spectral classes of this nature? (See problem 9.2). You may care to extend this discussion by considering how best to generate spectral classes for maximum likelihood, minimum distance and parallelepiped classification. This concept is discussed in J.A. Richards and D.J. Kelly, 1984: On the Concept of Spectral Class, Int. J. Remote Sensing, 5, 987–991.

**11.3** A maximum likelihood classifier can be developed using training data in the usual way by estimating class statistics. Describe how a threshold can be used to assist in the determination of the spectral class structure of the data.

**11.4** Spaceborne microwave remote sensing depends necessarily on the use of synthetic aperture radar (SAR) techniques. SAR images of agricultural regions display a substantial “speckle” owing to the coherent nature of the radiation employed. Comment on the effect speckle would have in trying to obtain accurate automated classification of agricultural radar images.

**11.5** This question relates to the effect of resampling on classification. Consider a single line of infrared image data, such as that corresponding, say, to Landsat 3 band 7 responses over a region that is vegetation to the left and water to the right. Imagine the vegetation/water boundary is sharp. Resample your single line of data onto a grid with the same centres as the original. However use both nearest neighbour and cubic convolution interpolation, the latter according to (2.11 a) with  $j' = 1$ . Comment on the results of classifying each of the

resampled lines of data given that the classifier could have been trained on classes that have infrared brightnesses between those of vegetation and water.

**11.6** Frequently texture is used as an element in the photointerpretation of airphotos or satellite images. It can be used, for example, in the discrimination of forested and grassy regions. When dealing with digital data using machine-assisted classified methods, texture can only be used if a means for computing the texture in the neighbourhood of a pixel can be determined. A simplistic measure is the standard deviation of pixel brightnesses in a  $3 \times 3$  neighbourhood about a pixel. Discuss how this texture measure can be incorporated into standard classification methods, noting any computational burdens involved.

**11.7** Sometimes the spectral domain for a particular sensor and scene does consists of a set of distinct clusters of data. As an illustration, a Landsat visible red versus near infrared two dimensional space of an image of a region of just water, sand and mangrove vegetation would appear to have three groups of pixels. More often than not however, especially for images of natural vegetated and soil regions, the spectral domain will be very much a continuum, owing to the different degrees of mixing of the various cover types that can occur in nature. One is then led to question the distinctness and uniqueness, not only of spectral classes, but information classes as well for regions such as these. In view of these remarks comment on the issues involved in the classification of natural regions both in terms of the definition of the set of information classes to be used and in terms of the training procedures to be employed.

**11.8** Manually design a simple decision tree that can be used efficiently with ETM+ data for classification into deep water, shallow water, green vegetation and soil.

**11.9** How effectively can canonical analysis be applied to image data with 200 spectral channels? Is the principal components transformation a viable alternative feature reduction procedure in this case?

**11.10** Discuss the concept of spectral class in relation to hyperspectral data.

## 12

# Multisource, Multisensor Methods

Frequently the need arises to analyse mixed spatial data bases, such as that depicted in Fig. 1.13. Such data sets could consist of satellite spectral, radar, hyperspectral, topographic and other point form data, all registered geometrically, as might be found in a geographic information system.

Labelling pixels by drawing inferences from several available sources of data is referred to as *data fusion* or *multisource classification*. As with the treatment of single image data sets, analysis of mixed data types can be carried out photointerpretatively or by using machine analysis.

Sometimes multisource classification is relatively straightforward, particularly if the different data sources are substantially of the same type and thus can be handled by the same sorts of photointerpreter knowledge or machine algorithm. In many cases, though, the problem is complex, especially when the analyst wishes to apply quantitative methods to data types that are quite different from each other. Manipulating satellite multispectral data with labelled map data is an example.

It is the purpose of this chapter to present some of the more common techniques for addressing the interpretation task quantitatively. Analysis by photointerpretation is not treated as such, since it depends largely on the analyst's skills with the range of data types present. Improving image quality for photointerpretative data fusion, however, is discussed by Gross and Schott (1997) and van der Meer (1997).

Numerically based quantitative methods are treated first, following which procedures based on evidential theory and expert systems are covered. The benefit of the latter is that the data sources do not need all to be in numerical form.

Clearly, the data to be analysed must first be registered. If the data has been retrieved from a geographic information system then that step will already have been performed. However, if spatial registration has not been carried out then the analyst will have to undertake that task using the procedures of Chap. 2. A word of caution is in order: the accuracy with which the interpretation of the mixed data can be performed will be influenced by the accuracy of the registration process in addition to the effectiveness of the analytical procedures (such as classification) employed.

## 12.1

### The Stacked Vector Approach

A straightforward way to classify mixed data is to form extended pixel vectors by stacking together the individual vectors that describe the various spectral and non-spectral data. This stacked vector will be of the form

$$X = [\mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_S^t]^t \quad (12.1)$$

where  $S$  is the total number of individual data sources with corresponding data vectors  $\mathbf{x}_1, \dots, \mathbf{x}_S$ , and the superscript “ $t$ ” denotes a vector transpose operation. The stacked vector  $X$  can, in principle, now be approached using standard classification techniques. This presents a number of difficulties if statistical methods such as maximum likelihood classification are considered. These include the incompatible statistics of the disparate data types, with some data unable to be represented by normal class models, and the quadratic cost increase with data dimensionality. Parallelepiped classification could be an appropriate algorithm to adopt since it depends only on the application of thresholds to components of the data vector  $X$ .

## 12.2

### Statistical Multisource Methods

#### 12.2.1

##### Joint Statistical Decision Rules

The single data source decision rule of (8.1) can be restated for multisource data described by (12.1) as

$$X \in \omega_i \quad \text{if } p(\omega_i|X) > p(\omega_j|X) \text{ for all } j \neq i$$

As with single source methods we can apply Bayes' theorem to give

$$X \in \omega_i \quad \text{if } p(X|\omega_i)p(\omega_i) > p(X|\omega_j)p(\omega_j) \text{ for all } j \neq i$$

To proceed further we need to find or estimate the class conditional joint source probabilities  $p(X|\omega_i) = p(\mathbf{x}_1, \dots, \mathbf{x}_S|\omega_i)$ . To render that exercise tractable independence among the data sources is generally assumed so that

$$p(X|\omega_i) = p(\mathbf{x}_1|\omega_i)p(\mathbf{x}_2|\omega_i)\dots p(\mathbf{x}_S|\omega_i)$$

where the  $p(\mathbf{x}_k|\omega_i)$  are the class conditional distribution functions derived from each data source individually. They are generally referred to as source specific class conditional density functions.

It is unlikely that the assumption of independence is valid but it is usually necessary in order to perform multisource statistical classification. With the assumption

the multisource decision rule can be written

$$\begin{aligned} X \in \omega_i \\ \text{if } p(\mathbf{x}_1|\omega_i) \dots p(\mathbf{x}_S|\omega_i) p(\omega_i) > p(\mathbf{x}_1|\omega_j) \dots p(\mathbf{x}_S|\omega_j) p(\omega_j) \\ \text{for all } j \neq i \end{aligned}$$

An important consideration with classification from multiple sources of data is whether each available data source has the same quality as far as the classification is concerned. Some data sets, for example, could be noisy and thus not contribute as well to the decision making process as other, well defined data sets. Just as a photointerpreter may qualify their judgement about particular data sets based on their visual quality when forming an opinion, we need to do that in the quantitative decision rule. That can be achieved by adding powers to the source specific class conditional probabilities to give

$$\begin{aligned} X \in \omega_i \\ \text{if } p(\mathbf{x}_1|\omega_i)^{\alpha_1} \dots p(\mathbf{x}_S|\omega_i)^{\alpha_S} p(\omega_i) > p(\mathbf{x}_1|\omega_j)^{\alpha_1} \dots p(\mathbf{x}_S|\omega_j)^{\alpha_S} p(\omega_j) \\ \text{for all } j \neq i \end{aligned}$$

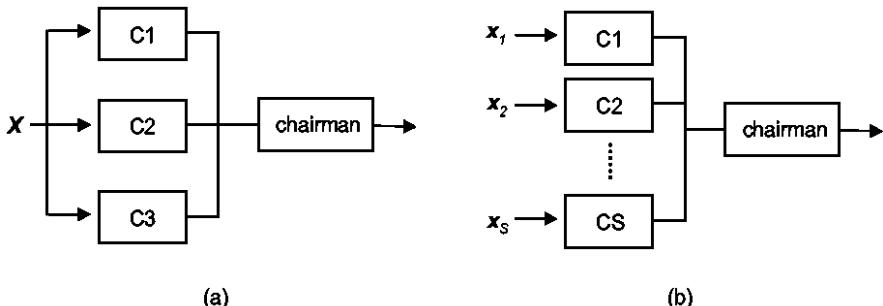
where the  $\alpha_s$  are a set of weighting factors chosen to enhance the influence of some sources (those most trusted) and to diminish the influence of other (perhaps the most noisy).

There are several problems with the joint statistical approach, in common with the stacked vector method of the previous section. First, each source must be able to be modelled to yield class conditional distribution functions. Secondly, the information classes must be consistent over the sources – in other words the set of information classes appropriate to one source (say multispectral) must be the same as those for the other sources (say radar and hyperspectral). This last requirement is a major limitation of multisource statistical methods.

## 12.2.2 Committee Classifiers

Closely related to the concept of handling the data sources independently is the concept of employing a set of individual classifiers, one operating on each data source. Sets of classifiers are usually referred to as committees, such as that seen in Sect. 8.9.3 and as illustrated more generally in Fig. 12.1. Note that it is a feature of committee classifiers that there is a chairman, whose role it is to consider the outputs of the individual classifiers and make a decision about the class membership of a pixel.

There are several logics that the chairman could use in decision making. One is the majority vote, in which the chairman decides the class most recommended by the committee members. Another is veto logic in which all the classifiers have to agree about the class membership of a pixel before the chairman will label the pixel. Yet another is seniority logic, in which the chairman always consults one particular classifier first (the most “senior”). If that classifier is able to recommend a class label



**Fig. 12.1.** **a** A committee of three classifiers in which each classifier sees all the data. **b** A committee in which each classifier is used to handle one of the data sources. C1 etc. are classifiers.

for a pixel then the chairman allocates that label. If the first (most senior) classifier is unable to make a reliable recommendation then the chairman consults the next most senior classifier, and so on until the pixel can be labelled.

Committee classifiers can be used in two ways. First, all the available data could be fed to all committee members so that each classifier in a sense handles a stacked vector, as depicted in Fig. 12.1a. Such an approach can be used also for single source analysis. The second way of using a committee on multisource data is to use one committee member per data source as shown in Fig. 12.1b. In this way each classifier can be optimised for handling one particular data type.

12.2.3

## Opinion Pools and Consensus Theoretic Methods

A variation on the committee classifier concept is the use of opinion pools. They depend upon finding the single source posterior probabilities and then combining them arithmetically or geometrically (logarithmically). The *linear opinion pool* computes a group membership function, similar to a joint posterior probability, of the form

$$f(\omega_i | X) = \sum_{s=1}^S \alpha_s p(\omega_i | x_s)$$

in which the  $\alpha_s$  are a set of weighting constants (which sum to unity) that control the relative influences of each source in the final value of the group membership function and thus in the labelling of the pixel. One limitation of this rule – known generally as a consensus rule – is that one data source tends to dominate the decision making (Benediktsson et al, 1997). Another acceptable consensus rule that doesn't suffer that limitation is the multiplicative version

$$f(\omega_i | \mathcal{X}) = \prod_{s=1}^S p(\omega_i | \mathbf{x}_s)^{\alpha_s}$$

which by taking the logarithm becomes the so-called logarithmic opinion pool consensus rule

$$\log\{f(\omega_i|X)\} = \sum_{s=1}^S \alpha_s \log\{p(\omega_i|\mathbf{x}_s)\}$$

Note that if one source posterior probability is zero then  $f(\omega_i|X) = 0$  and, irrespective of the recommendations from any of the other sources, the group recommendation is zero (before the log is taken) for that class-pixel combination. In other words one very weak source can veto a decision.

In the linear and logarithmic opinion pool rules the weighting coefficients  $\alpha_s$  again reflect the confidence we have in the respective data sets.

There may be cases though where one data source is better for some classes than the others, and likewise a different data source might be better for discriminating a different set of classes. It is possible therefore to choose values for the  $\alpha_s$  that will maximise the probability of a correct classification result in an average sense (Benediktsson et al, 1997).

## **12.2.4 Use of Prior Probability**

In the decision rule of (8.3) and discriminant function of (8.4) the prior probability terms tell us the probability with which the class membership of a pixel could be guessed based upon any information we have about that pixel prior to considering the available remotely sensed measurements. In its simplest form we assume it represents the relative abundance of that class in the scene being analysed. However, prior class membership can be obtained from other sources of information as well. In the case of the Markov Random Field approach to incorporating spatial context in Sect. 8.8.5, the prior term is the neighbourhood conditional prior probability.

Strahler (1980) and more recently Bruzzone et al (1997) have used the prior term in (8.4) to incorporate the effect of another data source – in Strahler’s case to bring the effect of topography into a multispectral classification of a forested region.

## **12.2.5 Supervised Label Relaxation**

The probabilistic label relaxation scheme in Sect. 8.8.4 can also be used to refine the results of a classification by bringing in the effect of another data source, while developing spatial neighbourhood consistency as well. The updating rule in (8.16) can have another step added to it for this purpose. Although heuristic in development it has been seen to perform well when embedding topographic data into a classification (Richards, Landgrebe and Swain, 1982).

Known as supervised relaxation, the updating rule at the  $k$ th iteration for class  $\omega_i$  on pixel  $m$  is

$$\begin{aligned} p_m^{k+1}(\omega_i)^* &= p_m^k(\omega_i) Q_m^k(\omega_i) && \text{for embedding spatial context} \\ p_m^{k+1}(\omega_i) &= p_m^{k+1}(\omega_i)^* \phi_m(\omega_i) && \text{for incorporating another data source} \end{aligned}$$

followed by application of (8.16), in which the denominator is a normalising factor. The term  $\phi_m(\omega_i)$  is the probability that  $\omega_i$  is the correct class for pixel  $m$  as far as another data source is concerned.

## 12.3

### The Theory of Evidence

A restriction with the previous methods for handling multisource data is that all the data must be in numerical form. Yet many of the data types encountered in a spatial data base are inherently non-numerical. The mathematical Theory of Evidence is a field in which the data sources are treated separately and their contributions combined to provide a joint inference concerning the correct label for a pixel, but does not, of itself, require the original data variables to be numerical. While it involves numerical manipulation of quantitative measures of evidence, the bridge between these measures and the original data is left largely to the user.

#### 12.3.1

##### The Concept of Evidential Mass

The essence of the technique involves the assignment of belief, represented as a so-called mass of evidence, to various labelling propositions for a pixel. The total mass of evidence available for allocation over the candidate labels for the pixel is unity. To see how this is done suppose a classification exercise, involving for the moment just a single source of image data, has to label pixels as belonging to one of just three classes:  $\omega_1$ ,  $\omega_2$  and  $\omega_3$ . It is important that the set of classes be exhaustive (i.e. cover all possibilities) so that  $\omega_3$  for example might be the class “other”. Suppose some means is available by which labels can be assigned to a pixel (which could include maximum likelihood methods if desired) which tells us that the three labels have likelihoods in the ratios 2 : 1 : 1. However, suppose we are a little uncertain about the labelling process or even the quality of the data itself, so that we are only willing to commit ourselves to classifying the pixel with about 80% confidence. Thus we are about 20% uncertain about the labellings, even though we are reasonably happy about the relative likelihoods. Using the symbolism of the Theory of Evidence, the distribution of the unit mass of evidence over the three possible labels, and our uncertainty about the labelling, is expressed:

$$m((\omega_1, \omega_2, \omega_3, \theta)) = \langle 0.4, 0.2, 0.2, 0.2 \rangle \quad (12.2)$$

where the symbol  $\theta$  is used to signify the uncertainty in the labelling<sup>1</sup>. Thus the mass of evidence assigned to label  $\omega_1$  as being correct for the pixel is 0.4, etc. (Note that if we were using straight maximum likelihood classification, without accounting for uncertainty, the probability that  $\omega_1$  is the correct class for the pixel would have been 0.5). We now define two further evidential measures. First, the *support* for a labelling proposition is the sum of the mass assigned to the proposition and any of its subsets. Subsets are considered later. The *plausibility* of the proposition is one minus the total support of any contradictory propositions. Support is considered to be the minimum amount of evidence in favour of a particular labelling for a pixel whereas plausibility is the maximum possible evidence in favour of the labelling. The difference between the measures of plausibility and support is called the *evidential interval*; the true likelihood that the label under consideration is correct for the pixel is assumed to lie somewhere in that interval. For the above example, the supports, plausibilities and evidential intervals are:

$$\begin{array}{lll} s(\omega_1) = 0.4 & p(\omega_1) = 0.6 & p(\omega_1) - s(\omega_1) = 0.2 \\ s(\omega_2) = 0.2 & p(\omega_2) = 0.4 & p(\omega_2) - s(\omega_2) = 0.2 \\ s(\omega_3) = 0.2 & p(\omega_3) = 0.4 & p(\omega_3) - s(\omega_3) = 0.2 \end{array}$$

In this simple case the evidential intervals for all labelling propositions are the same and equal to the mass allocated to the uncertainty in the process or data as discussed above, i.e.  $m(\theta) = 0.2$ . Consider another example involving four possible spectral classes for the pixel, one of which represents our belief that the pixel is in either of two classes. This will demonstrate that, in general, the evidential interval is different from the mass allocated to uncertainty. Suppose the mass distribution is:

$$m(\langle \omega_1, \omega_2, \omega_1 \vee \omega_2, \omega_3, \theta \rangle) = \langle 0.35, 0.15, 0.05, 0.3, 0.15 \rangle$$

where  $\omega_1 \vee \omega_2$  represents ambiguity in the sense that, for the pixel under consideration, while we are prepared to allocate 0.35 mass to the proposition that it belongs to class  $\omega_1$  and 0.15 mass that it belongs to class  $\omega_2$ , we are prepared also to allocate some additional mass to the fact that it belongs to either of those two classes and not any others.

For this example the support for  $\omega_1$  is 0.35 (being the mass attributed to it) whereas the plausibility that  $\omega_1$  is the correct class for the pixel is one minus the support for the contradictory propositions. There are two – i.e.  $\omega_2$  and  $\omega_3$ . Thus the plausibility of  $\omega_1$  is 0.55, and the corresponding evidential interval is 0.2 (different now from the mass attributed to uncertainty). The support given to the mixture class  $\omega_1 \vee \omega_2$  is 0.55, being the sum of masses attributed to that class and its subsets.

To see how the Theory of Evidence is able to cope with the problem of multisource data, return now to the simple example given by the mass distribution in (12.2). Suppose there is available a second data source which is also able to be labelled

---

<sup>1</sup> Strictly, in the Theory of Evidence,  $\theta$  represents the set of all possible labels. The mass associated with uncertainty has to be allocated somewhere; thus it is allocated to the set as a whole.

into the same set of spectral classes. Again, however, there will be some uncertainty in the labelling process which can be represented by a measure of uncertainty as before; also, for each pixel there will be a set of likelihoods for each possible label. For a particular pixel suppose the mass distribution after analysing the second data source is

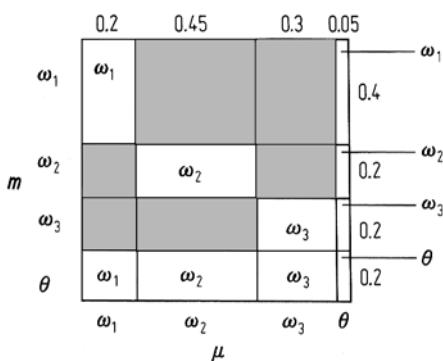
$$\mu((\omega_1, \omega_2, \omega_3, \theta)) = \langle 0.2, 0.45, 0.3, 0.05 \rangle \quad (12.3)$$

Thus, the second analysis seems to be favouring  $\omega_2$  as the correct label for the pixel, whereas the first data source favours  $\omega_1$ . The Theory of Evidence now allows the two mass distributions to be merged in order to combine the evidences and thus come up with a label which is jointly preferred and for which the overall uncertainty should be reduced. This is done through the mechanism of the orthogonal sum.

### 12.3.2

#### Combining Evidence – the Orthogonal Sum

Dempster's orthogonal sum is illustrated in Fig. 12.2. It is performed by constructing a unit square and partitioning it vertically in proportion to the mass distribution from one source and horizontally in proportion to the mass distribution from the other source. The areas of the rectangles thus formed are calculated. One rectangle is formed from the masses attributed to uncertainty ( $\theta$ ) in both sources; this is considered to be the remaining uncertainty in the labelling process after the evidences from both sources have been combined. Rectangles formed from the masses attributed to the same class have their resultant (area) mass assigned to that class. Rectangles formed from the product of mass assigned to a particular class in one source and mass assigned to uncertainty in another source have their resultant mass attributed to the specific class. Similarly, rectangles formed from the product of a specific label, say  $\omega_2$  and an ambiguity, say  $\omega_1 \vee \omega_2$ , are allocated to the specific class. Rectangles formed from different classes in the two sources are contradictory and are not used in computing merged evidence. In order that the resulting mass distribution sums to unity a normalising denominator is computed as the sum of the areas of all the rectangles that are not contradictory. For the current example this factor is 0.47. Thus,



**Fig. 12.2.** Graphical illustration of the application of the Dempster orthogonal sum for merging the evidences from two data sources; the labels in the white squares indicate the class to which the mass is attributed

after the orthogonal sum has been computed the resulting (combined evidence) mass distribution is:

$$m(\omega_1) = (0.08 + 0.02 + 0.04)/0.47 = 0.298$$

$$m(\omega_2) = (0.09 + 0.01 + 0.09)/0.47 = 0.404$$

$$m(\omega_3) = (0.06 + 0.01 + 0.06)/0.47 = 0.277$$

$$m(\theta) = 0.01/0.47 = 0.021$$

Thus class 2 is seen to be recommended jointly. The reason for this is that source 2 favoured class 2 and had less uncertainty. While source 1 favoured class 1, its higher level of uncertainty meant that it was not as significant in influencing the final outcome.

The orthogonal sum can also be expressed in algebraic form (Lee et al. 1987, Garvey et al. 1981). If two mass distributions are denoted  $m_1$  and  $m_2$  then their orthogonal sum is:

$$m_{12}(z) = \mathcal{K} \sum_{(x \cap y = z)} m_1(x) \cdot m_2(y) \triangleq m_1(x) \oplus m_2(x)$$

where

$$\mathcal{K}^{-1} = \sum_{(x \cap y \neq \varphi)} m_1(x) \cdot m_2(y)$$

in which  $\varphi$  is the null set. In applying these formulas it is important to recognise that

$$(x \vee y) \cap y = y$$

$$\theta \cap y = y$$

For more than two sources, the orthogonal sum can be applied repetitively since the expression is both commutative (the order in which the sources are considered is not important) and associative (can be applied to any pair of sources and then a third source, or equivalently can be applied to a different pair and a third source).

### 12.3.3 Decision Rule

After the orthogonal sum has been applied the user can then compute the support for and plausibility of each possible ground cover class for a pixel. Two following steps are then possible. First a decision rule might be applied in order to generate a single thematic map in the same manner as is done with statistical classification methods.

A number of candidate decision rules are possible including a comparison of the supports for the various candidate classes and a comparison of plausibilities as discussed in Lee et al. (1987). Generally a maximum support decision rule would be used, although if the plausibility of the second most favoured class is higher than

the support for the preferred label, the decision must be regarded as having a degree of risk.

Secondly, rather than produce a single thematic map, it is possible to produce a map for each category showing the distribution of supports (or plausibilities). This might be particularly appropriate in a situation where the ground cover classes are not well resolved (such as in a geological classification, for an illustration of which see Moon (1990)).

## 12.4 Knowledge-Based Image Analysis

Techniques for the analysis of mixed data types, such as the multisource statistical classification and evidential methods treated above, have their limitations. Apart from their complexities, most are restricted to data that is inherently in numerical form, such as that from multispectral and radar imaging devices, along with quantifiable terrain data like digital elevation maps. Yet, in the image data base of a Geographic Information System (GIS), for example, there are many spatial data sets that are non-numerical but which would enhance considerably the results expected from an analysis of a given geographical region if they could be readily incorporated into the decision process. These include geology and soil maps, planning maps and even maps showing power, water and road networks. It is clear therefore that quite a different approach for handling non-numerical data is required, particularly when a user wishes to exploit the richness of information imbedded in the multisource, multisensor data environment of a GIS. The Theory of Evidence treated in Sect. 12.3 is one possibility, but it still requires the analysis task to be expressed in a quantifiable form so that numerical manipulation of evidence is possible. To avoid having to establish this bridge, a method for *qualitative* reasoning would be a particular value.

The adoption of expert systems or knowledge-based methods offers promise in this regard. It is the role of this section to outline some of the fundamental aspects of such processes and to demonstrate their potential. The field is very diverse and, as will become clear in reading the following, the use of one particular approach may be guided by individual preferences and available software rather than a perception of what is the most appropriate algorithm for a given purpose. What will become clear however is that the use of (often qualitative) interpreter knowledge greatly aids analysis; moreover, quite simple knowledge-based methods can yield surprisingly good results.

### 12.4.1 Knowledge Processing: Emulating Photointerpretation

To develop the theme of a knowledge-based approach it is of value to return to the comparison of the attributes of photointerpretation and quantitative analysis developed in Table 3.1. However, rather than making the comparison solely on the basis

of a single source of multispectral data, as was the case in Chapter 3, consider now that the data to be analysed consists of three parts: a Landsat multispectral image, a radar image of the same region and a soil map of that region. From what has been said above, standard methods of quantitative analysis cannot cope with trying to draw inferences about the cover types in the region since they will not function well on two numerical sources of quite different characteristics (multispectral and radar data) and also since they cannot handle non-numerical data at all.

In contrast, consider how a skilled photointerpreter might approach the problem of analysing this multiple source of spatial data. Certainly he or she would not wish to work at the individual pixel level, as discussed in Sect. 3.1, but would more likely concentrate on regions. Suppose a particular region was observed to have a predominantly pink tone on a standard false colour composite print of the multispectral data, leading the photointerpreter to infer initially that the region is vegetated; whether it is a grassland, crop or forest region may not yet be clear. However the photointerpreter could then refer to the radar imagery. If its tone is dark, then the region would be thought to be almost smooth at the radar wavelength being used. Combining this evidence with that from the multispectral source, the photointerpreter is then led to consider the region as being either grassland or a small crop. He or she might then resolve this conflict by referring to the soil map of the region. Noting that the soil type is not that normally associated with agriculture, the photointerpreter would then conclude that the region is same form of natural grassland.

In practice the process of course may not be so straightforward, and the photointerpreter may need to refer backwards and forwards over the data sets in order to finalise an interpretation, especially if the multispectral and radar tones were not uniform for the region. For example, some spots on the radar imagery may be bright. The photointerpreter would probably regard these as indicating shrubs or trees, consistent with the overall region being labelled as natural grassland. The photointerpreter will also account for differences in data quality, placing most reliance on data that is seen to be most accurate or most relevant to a particular exercise, and weighting down unreliable or marginally relevant data.

The question we need to ask at this stage is how the photointerpreter is able to make these inferences so easily. Even apart from spatial processing, as discussed in Table 3.1 (where the photointerpreter would also use spatial clues such as shape and texture), the key to the photointerpreter's success lies in his or her *knowledge* – knowledge about spectral reflectance characteristics, knowledge of radar response and also of how to combine the information from two or more sources (for example, pink multispectral appearance *and* dark radar tone indicates a low level vegetation type). We are led therefore to consider whether the knowledge possessed by an expert such as a skilled photointerpreter can be given to and used by a machine and so devise a method for analysis that is able to handle the varieties of spatial data type available in GIS-like systems. In other words can we emulate the photointerpreter's approach? If we can then we will have available an analytical procedure capable of handling mixed data types, and also able to work repetitively, at the pixel level if necessary. With respect to the latter point, it is important to recognise that photointerpreters generally

work at a regional rather than a pixel level; knowledge-based image analysis is able to follow such an approach if segments in image data have previously been identified using region growing techniques such as that used in ECHO (Sect. 8.8.2).

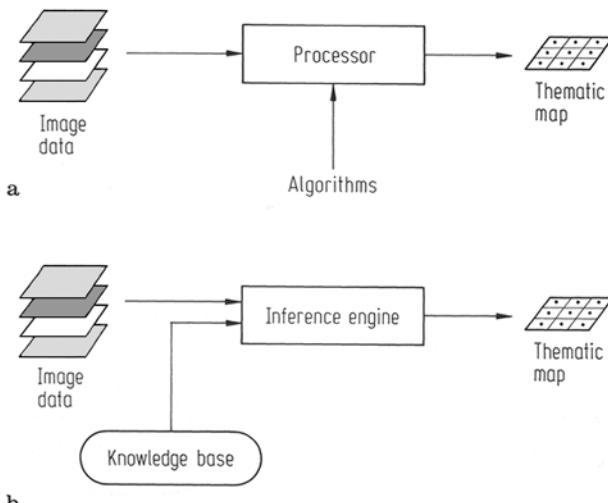
### 12.4.2

## Fundamentals of a Knowledge-Based Image Analysis System

### 12.4.2.1 Structure

If we were to visualise the structure of a traditional supervised classification approach to the analysis of image data we might come up with the block diagram shown in Fig. 12.3a. The data to be analysed is fed to a processor (computer) which is also supplied with the algorithms (maximum likelihood rule, minimum distance rule etc.) appropriate to the task. The algorithms are applied pixel by pixel to produce a label for each pixel, dependent solely on the class signatures and the characteristics of the data. It can be argued that some expert knowledge has been supplied to the process by the user in relation to the selection of algorithm to use and, more particularly, in selecting the reference data with which to train the classifier. The user, however, need not possess any detailed knowledge of spectral reflectance characteristics or other properties in order that the analysis proceed and results be produced. As we have seen in previous chapters quite good results can be achieved provided only that training regions are chosen carefully and any multimodelling is removed when using maximum likelihood classification.

In contrast, Fig. 12.3b shows the structure of a knowledge-based approach to the analysis. Again, the spatial data to be analysed is fed to the processor, but so is a



**Fig. 12.3.** **a** Traditional image analysis computing. **b** Knowledge based image analysis system.

knowledge base. The knowledge stored in this knowledge base has been obtained from experts in the field of the analysis and stored in such a manner (see Sect. 12.4.2.2) that it can be used to analyse the data. The knowledge is applied to the data in the processor by what is called an inference mechanism, or sometimes an inference engine. Its role is to interpret the knowledge base, apply the knowledge to the data, and make, and keep track of, decisions about the class memberships of pixels.

### **12.4.2.2 Representation of Knowledge: Rules**

There are several ways in which expert knowledge can be captured and recorded for use by a knowledge-based analysis system (Sell, 1985; Frost, 1986). The simplest, and perhaps most common, is to use rules (sometimes called production rules). These are of the form:

**if condition then inference.**

'Condition' in the rule is a logical expression which can be either true or false. If it is true then the inference is justified otherwise no information is provided by that rule. 'Condition' can be a *simple* logical expression or can be a *compound* logical statement in which several components are linked through the logical **or** and **and** operations. These operations are defined as:

The composite condition (condition 1 **and** condition 2) is true only if condition 1 and condition 2 are **both** true.

The composite condition (condition 1 **or** condition 2) is true if **either** condition 1 **or** condition 2 is true.

Note that rule-based knowledge systems can also make use of the logical **not** operation, defined by:

**not** (condition) is false if condition is true, and vice versa.

Each single rule can be thought of as encapsulating one item of knowledge. For example, a rule which could be applied to a Landsat MSS pixel to check whether it is likely to be vegetated might be:

**if** near infrared response > red response **then** vegetation.

Although this is a weak rule, we know it is correct from our knowledge of the spectral reflectance characteristics of vegetation. Similarly, a rule that would reveal a region to be a smooth (specular or near specular) surface could be:

**if** radar tone is dark **then** smooth surface.

Note that these rules need not be conclusive, but rather they should simply provide a degree of evidence in favour of pixels having the labels specified. Sometimes the knowledge contained in several rules might be necessary to enable a pixel to be identified with any degree of certainty.

A knowledge base in such an analysis system might contain many hundreds of rules of these types, obtained from experts in particular fields. When image data is presented to the inference engine for analysis, the engine goes through the rule base checking the support for or against various labelling propositions. Some rules will offer strong support while others will be weak, as illustrated above. Also, several candidate classes for a particular pixel might find support among the rules; procedures are then required for resolving among them. Possible means for doing this are described in the following sections.

As an example of a simple rule representation of knowledge, suppose a particular Landsat MSS image has to be segmented into just vegetation, water and other (unspecified) cover types. The following set of rules should be able to accomplish this task:

```

if band 7/band 5 > threshold then vegetation
if band 7/band 4 < 1 then water
if not (water) and not (vegetation) then other

```

Notice that the third rule supposes for this particular exercise that anything that is not water or vegetation must be other. Also note that this rule has two conditions (sometimes called antecedents) that are logically ‘and-ed’. Both must be true in order that the total antecedent is true and thus the inference (sometimes called the consequent) is justified. In the first rule a parameter is used – i.e. ‘threshold’. This requires a numerical value to be available, which will almost certainly be scene dependent. The value could be provided to the system before the analysis starts by the user entering it manually or, alternatively, a small training region of vegetation could be used from which the value could be learnt. Many of the rules encountered in remote sensing image analysis will require parameters such as thresholds.

The rules illustrated here, and indeed most of those to be encountered in this treatment of knowledge-based methods, rely on spectral or similar pixel-specific knowledge. In many expert systems devised for the analysis of remote sensing and GIS data, spatial constraints are also used as a source of knowledge and appropriate rules are developed (Ton et al., 1991). Even spectrally derived rules may not rely on simple expressions and comparisons of bands. Spectral contrasts, such as the brightness in a given band compared with total image brightness, can also be used (Wharton, 1987).

### **12.4.2.3**

#### **The Inference Mechanism**

The inference engine or mechanism can be quite simple if the knowledge-based system is very specific to a particular application, or can be more complex and powerful if a general expert system is required. In the simple example of the previous section all the inference mechanism has to do is to check which of the rules gives a positive response for each pixel in the image and then label the pixel accordingly. More generally, however, when large rule sets are used, the inference mechanism needs to keep track of all the rules that infer a particular cover type, along with

those that infer that the pixel is not of that cover type and, similarly, the rules that suggest the pixel is or is not from other candidate classes; finally it has to make a decision about the correct class by weighing all the evidence from the rules. It may also have to account for redundant reasoning and circular arguments, and has to be able to assess whether long reasoning chains carry as much weight in the decision process as inferences that might involve only a single decision in coming up with the label for a pixel. In addition, an effective inference process will also allow uncertainties in data quality, missing data and missing rules to be accommodated. That degree of complexity is beyond this introductory treatment; a full discussion of all of these issues will be found in Srinivasan (1990) and Srinivasan and Richards (1993). However, it is of value to consider briefly something of the complexity that can be built into the inference mechanism in order to emulate more closely the reasoning process that might be adopted by a typical photointerpreter. To do this, it is instructive first to consider the approach used by Wharton (1987).

Wharton uses eight bands of Thematic Mapper Simulator data, centered on:

band 1	0.485 $\mu\text{m}$	band 2	0.560 $\mu\text{m}$
band 3	0.660 $\mu\text{m}$	band 4	0.880 $\mu\text{m}$
band 5	1.150 $\mu\text{m}$	band 6	1.650 $\mu\text{m}$
band 7	2.215 $\mu\text{m}$	band 8	11.400 $\mu\text{m}$

He then establishes spectral rule sets for each of his classes of interest. These rules are in three groups, one of which compares band combinations. For example, a rule for determining whether a pixel might belong to the green vegetation class is:

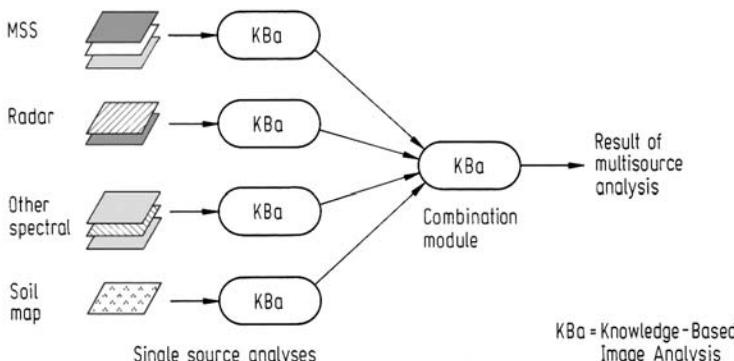
**if** average of bands 4 and 5 > sum of bands 2 and 3  
**then** class is green vegetation [5, 20]

The figures in square brackets are measures of evidence in favour of and against the labelling proposition. Thus, when testing for the green vegetation category, if the test is positive then the evidence in support of this being the correct class for the pixel is incremented by 5. If the test fails then the evidence against the class for the pixel being green vegetation is incremented by 20. It is the function of the inference process to keep track of all the evidence in favour of and against each class for a given pixel and then, when all the rules have been used, to test the accumulated evidence and decide the most appropriate class for the pixel from the perspective of spectral data.

### 12.4.3

#### Handling Multisource and Multisensor Data

There are two approaches that might be adopted when considering the development of a knowledge-based approach for the analysis of data that comes from more than one source or sensor. If the analysis is strongly focussed on a particular application it might be appropriate to consider a single knowledge base which contains all the rules, including those rules necessary to process two or more sources together. As a simple illustration, the following rule would be used to determine if a particular



**Fig. 12.4.** Decomposed multisource analysis using a knowledge based approach

region might be urban if both radar and multispectral data were available:

**if red response is high and radar tone is high then urban**

Of course, use of this rule requires a specification of what ‘high’ means for both the multispectral and radar data. However, given that those thresholds are available, rules such as this can be used to process the data sources jointly.

Possibly a more practical approach is, first, to decompose the multisource, multisensor problem into a set of individual analyses and then combine their results in a separate expert system that is able to perform the joint analysis as depicted in Fig. 12.4. Each individual analysis module and the combination module will have its own rule base and inference mechanism. The advantages of this approach are that the rule sets are each focussed on a particular sensor and that results can be updated at a later time if and when new data sources become available. This is a particularly important consideration in the context of a GIS.

Separate knowledge bases to be used for a simple segmentation could be:

For an MSS source:

**if band 7/band 5 > 3.0 then vegetation**  
**if band 7/band 4 < 0.9 then water**  
**if (band 4 + band 5)/(band 6 + band 7) > 0.6 then soil**

For the radar source (see Fig. 1.5):

**if radar tone < threshold 1 then specular surface**  
**if radar tone > threshold 2 then corner reflector effect**  
**if threshold 1 < radar tone < threshold 2 then diffuse surface or volume scattering**

On the basis of these rules it is presumed we are not able to discriminate diffuse surface scattering and volume scattering. The combined inference ‘diffuse surface or volume scattering’ is thus the best that can be done, where appropriate, in the following.

No separate rules are to be used at this stage for the soil map in Fig. 12.4, since it already consists of a set of labels for each pixel.

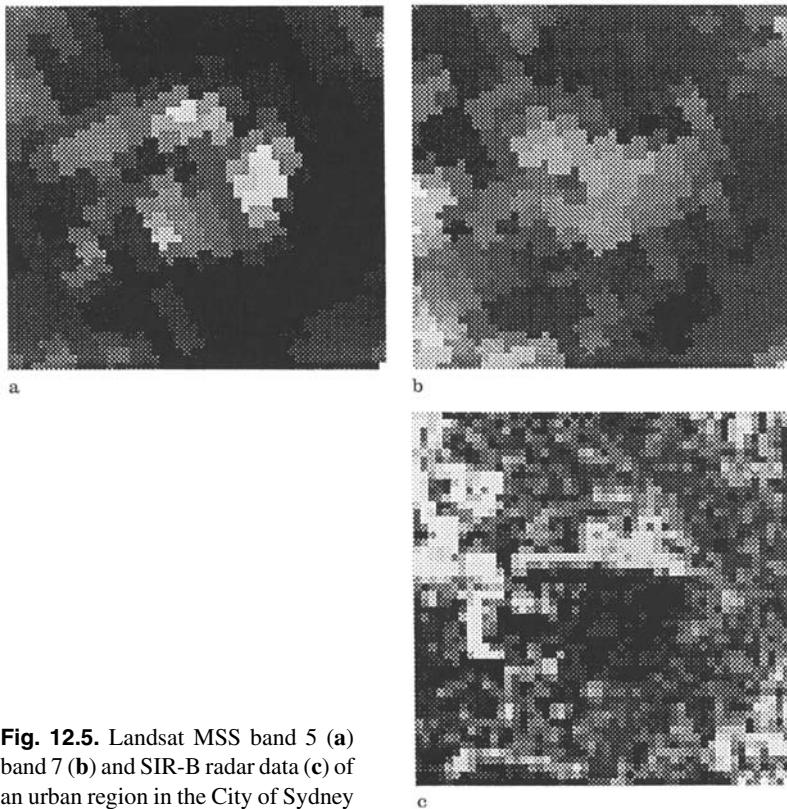
If the rules sets above are applied, as appropriate, to the MSS and radar data we will produce labels for each pixel from the individual analyses. For example the MSS data might specify a pixel as vegetation and the radar data classify it as a specular surface, while the soil map indicates a loam soil type. What the combination knowledge-base has to do is to process these labels to come up with a specific land cover category for the pixel, of the type needed by the user. A set of rules that might be found in the combination module therefore could be:

- if soil and specular surface then bare ground**
- if soil and corner reflector effect then urban**
- if vegetation and specular surface then low level vegetation**
- if vegetation and diffuse surface or volume then trees or shrubs**
- if low level vegetation and loam then crops**
- if low level vegetation and sand then grassland**
- if low level vegetation and clay then grassland**
- if water and specular surface then lake**
- if water and diffuse surface or volume then open water**

Whereas all previous examples of rules have had numerical conditions to test, these combination rules have conditions defined in terms of labels. This decomposition strategy is illustrated in the following example.

#### 12.4.4 An Example

Figure 12.5 shows Landsat MSS bands 5 and 7 and an L band SIR-B synthetic aperture radar image for a small urban area in Sydney's north-western suburbs. The Landsat data is unable to distinguish between urban areas and areas cleared for development. The radar data on the other hand, provides structural information, but no information on the actual cover type. The knowledge-based analysis system developed by Srinivasan and Richards (1993) is able to analyse the images jointly and thus develop a cover type map that resolves classes that are confused in either the Landsat or radar data alone. Full details of this and other applications of this approach will be found in Srinivasan (1990). In the following sections a summary of the expert system used is provided. It is based upon a decomposition philosophy of the style shown in Fig. 12.4 but, in its full version, also has a final module that allows spatial knowledge to be applied to the output of the combination module. The latter is an important component of photointerpretation and can be handled in knowledge-based analysis by region growing beforehand or by applying neighbourhood relations during or after analysis. For simplicity in this example, only a pixel-based approach is discussed.



**Fig. 12.5.** Landsat MSS band 5 (a) band 7 (b) and SIR-B radar data (c) of an urban region in the City of Sydney

#### 12.4.4.1 Rules as Justifiers for a Labelling Proposition

In this method, production rules of the form outlined above are referred to as *justifiers* since they provide a degree of justification or evidence in favour of a particular labelling proposition. Expressing a rule in its generic form:

**if** condition **then** inference

the approach specifies four types of rule:

Conclusive      If the condition is true then the justification for the inference is conclusive (i.e. absolute).

For example:

**if** radar tone is black **then** radar shadow.

Prima Facie      If the condition is true then there is reason to believe that the inference is true. If the condition is false it cannot be concluded in general that the inference is false.

For example:

**if** MSS band 7/MSS band 5 > 2 **then** vegetation.

Criterion

This is a special *prima facie* justifier for which a false condition provides *prima facie* justification to disbelieve the inference.

For example:

**if** MSS band 7 < MSS band 4 **then** water

(noting that if MSS band 7 > MSS band 4 then definitely not water).

Contingent

If the condition is true then support is provided for other, *prima facie*, reasons to believe the inference. These types of rule are not sufficient in themselves to justify the inference.

For example:

**if** MSS band 7 > MSS band 5 **then** vegetation.

This structuring of justifications is not unlike the strengths of reasoning used by photointerpreters. In some cases the evidence would suggest to a photointerpreter that the cover type simply *must* be of a particular type. In other cases the evidence might be so slight as simply to suggest what the cover type might be – indeed the photointerpreter might even withhold making a decision in such a situation until some further evidence is available.

This has been a simple review of the concept of justifiers in qualitative reasoning systems. A fuller treatment, which considers justifiers as inferences in a so-called defeasible logic, can be found in Nute (1988), Pollock (1974) and Srinivasan (1990).

#### 12.4.4.2

#### Endorsement of a Labelling Proposition

The justifiers of the previous section play a major role in reasoning. At any given stage in the reasoning process an inference may have valid reasons for and against it. It is then necessary to resolve among these to determine the most supported label. This is the role of the endorsement

The endorsement of a label is the final level of justification for an inference. Given a set of justifiers for and against an inference, the implementation used in this example of a scene interpretation system employs the following endorsements:

The inference **is Definitely True** if there is at least one conclusive justifier in support.

The inference **is Likely To be True** if there is some net *prima facie* evidence in support.

The inference <b>is Indicated</b>	if, in the absence of <i>prima facie</i> justification, there are some net contingent justifiers in its favour.
A proposition <b>is Null</b>	if all justifiers for the belief are balanced by those for opposing beliefs.
A proposition <b>is Contradicted</b>	if it has conclusive justifiers balanced for and against it.
A labelling proposition is said to be <b>Unknown</b> if nothing is known about it.	
Complements of these endorsements also exist.	

After all the rules in the knowledge base have been applied to a pixel under examination, each of the possible labels will have some level of endorsement. That with the strongest endorsement is chosen as the label most appropriate for the pixel. Endorsements for other labels, although weaker, may still have value: for example, the two endorsements for a pixel that ‘grassland is likely to be true’ and ‘soil is indicated’ are fully consistent – the cover type may in fact be a sparse grassland, which the analyst would infer from the pair of endorsements.

If an endorsement falls in the last three categories above the pixel would be left unclassified.

#### 12.4.4.3 Knowledge Base and Results

The knowledge base for this exercise consisted of the following rules (Srinivasan, 1990).

For the Landsat MSS data source:

<b>if</b> band 7/band 5 is approximately 1	<b>then</b> contingent support for urban <b>and</b> contingent support for soil
<b>if</b> band 7/band 5 is moderate	<b>then</b> contingent support for urban, <b>and</b> contingent support for vegetation
<b>if</b> band 7/band 5 is high	<b>then</b> <i>prima facie</i> support for vegetation

These rules have to be trained in order to establish what is meant by moderate and high.

For the SIR-B data source:

<b>if</b> radar response is low	<b>then</b> <i>prima facie</i> support for specular behaviour
---------------------------------	---

**if** radar response is moderate      **then** prima facie support for volume scattering

**if** radar response is high      **then** prima facie support for corner reflector

Similarly the thresholds between low, moderate and high are established using small training areas.

The combination rules used by Srinivasan are:

**if** vegetation is likely to be true **and**  
corner reflector is likely to be true **then** prima facie  
support for woody vegetation

**if** vegetation is likely to be true **and**  
volume scattering is likely to be true **then** prima facie  
support for vegetation

**if** vegetation is likely to be true **and** specular  
behaviour is likely to be true **then** prima facie  
support for grassland

**if** soil is likely to be true **and** specular behaviour  
is likely to be true **then** prima facie  
support for cleared land

**if** vegetation is indicated **and** corner  
reflector is likely to be true **then** prima facie  
support for residential

**if** vegetation is indicated **and** volume  
scattering is likely to be true **then** prima facie  
support for residential

**if** urban is likely to be true **and** corner  
reflector is likely to be true **then** prima facie  
support for buildings

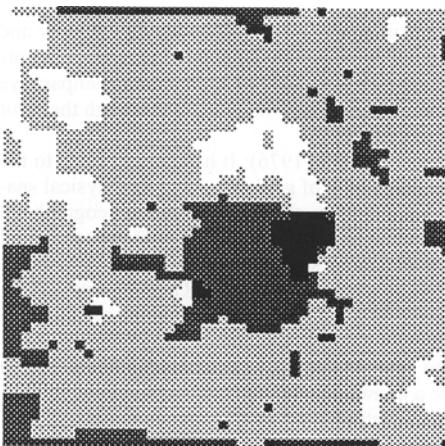
**if** vegetation is indicated **and** vegetation is not likely to be true  
**and** specular behaviour  
**then** contingent support for grassland

Note that the conditions tested in these roles are endorsements from the single source knowledge-base analyses.

Applying these rules yields the thematic map of Fig. 12.6, while Table 12.1 summarises the results quantitatively, using a careful photointerpretation of the data sets, and local knowledge, to provide the necessary ground truth data. Figure 12.6 demonstrates that the classifier is able to distinguish between grasslands and woody vegetation, owing to the structural information present in the radar image. Note also that not all bright regions in the radar image are classified as urban. Some actually

**Table 12.1.** Results of Combined Multispectral and Radar Analysis  
Overall accuracy = 77.3% (area weighted = 81.5%)

Ground Classes	Classes Identified by Rule-based Classification						
	Cleared land	Grassland (likely)	Grassland (indicated)	Woody vegetation	Residential	Buildings	Soil
Cleared land	82.5	2.5	5.0	2.5	0	0	7.5
Grassland	2.5	57.2	20.8	16.1	2.1	0.7	0
Woody vegetation	0	6.6	0	88.1	5.2	0	0
Urban	0	0.8	3.5	13.5	70.6	10.7	0.9



**Fig. 12.6.** Thematic map produced by knowledge based analysis of the data in Fig. 12.5. Classes are: black = soil, dark grey = grassland, light grey = woody vegetation, white = urban (cleared land, buildings, residential)

correspond to rows of trees; the confusion has been resolved using the land-cover information present in the Landsat image.

## References for Chapter 12

Schistad Solberg et al. (1994) develop the multisource statistical method of Sect. 12.2 further and provide means by which joint decisions can be made from multiple sources of data, while incorporating the effect of reliability of the data sources. Bruzzone et al. (1997) provide a comparative study of maximum likelihood methods (modified to bring in ancillary information through the prior probabilities) and neural networks for multisource classification.

Full details of the Theory of Evidence can be found in Shafer (1976). It has been applied to the problem of image analysis by Lee et al. (1987) and to integration of geological and geophysical spatial data sets by Moon (1990) while Garvey (1987) has used the method for describing geographical areas. Gong (1996) considers both the Theory of Evidence and the application of feed forward neural networks as methodologies for data fusion involving integrated spatial data types. Peddle (1995a) has incorporated evidential reasoning into a software scheme called MERCURY $\oplus$  as a means for multisource classification. Peddle (1995b) has

also discussed how measures of evidence can be generated from histograms of class training data.

The application of knowledge-based techniques to remote sensing was demonstrated by Nagao and Matsuyama (1980). Carlotto et al. (1984) describe a knowledge-based classification system for a single source of data, as does Mulder et al. (1988). A spectral rule-based approach for urban land cover discrimination using Landsat TM has been demonstrated by Wharton (1987), while Ton et al. (1991) demonstrate the use of both spectral and spatial knowledge for segmentation of Landsat imagery. Nicolin and Gabler (1987) describe a system for automatic interpretation of suburban scenes while Goldberg et al. (1985) describe a multi-level expert system for updating forestry maps with Landsat data, that has led to the development of a general purpose shell (Goodenough et al., 1987). Schowengerdt (1989) describes a system which enables inexperienced users perform rule-based image processing tasks. Kartikay et al. (1995) demonstrate the application of rule based expert systems to land cover analysis. Duch et al. (2004) provide a good overview of rule-based methods in general.

Knowledge of the radar response of terrain at different angles of incidence is used by Dobson et al. (1996) to develop a knowledge based approach to (structural) land cover classification from two radar sensors (ERS-1 and JERS-1). Solaiman et al. (1998) show how fusion of thematic map and edge information, both obtained from the same image data, can be used to improve a final map product.

- J.A. Benediktsson, J.R. Sveinsson and P.H. Swain, 1997: Hybrid Consensus Theoretic Classification. *IEEE Trans. Geoscience and Remote Sensing*, 35, 833–843.
- L. Bruzzone, C. Conese, F. Maselli and F. Roli, 1997: Multisource Classification of Complex Rural Areas by Statistical and Neural-Network Approaches. *Photogrammetric Engineering and Remote Sensing*, 63, 523–533.
- M.J. Carlotto, V.T. Tom, P.W. Baim and R.A. Upton, 1984: Knowledge-Based Multispectral Image Classification. *SPIE Vol. 504, Applications of Digital Image Processing VII*, 45–53.
- M.C. Dobson, L.E. Pierce and F.T. Ulaby, 1996: Knowledge-Based Land-Cover Classification Using ERS-1/JERS-1 SAR Composites. *IEEE Trans Geoscience and Remote Sensing*, 34, 83–99.
- W. Duch, R. Setiono and J.M. Zurada, 2004: Computational Intelligence Methods for Rule-Based Data Understanding. *Proc. IEEE*, 92, 771–805.
- R. Frost, 1986: Introduction to Knowledge Base Systems, McGraw-Hill, New York.
- T.D. Garvey, 1987: Evidential Reasoning for Geographic Evaluation for Helicopter Route Planning. *IEEE Trans Geoscience and Remote Sensing*, GE-25, 294–304.
- T.D. Garvey, J.D. Lowrance and M.A. Fisher, 1981: An Inference Technique for Integrating Knowledge from Disparate-Sources. *Proc 7th Int. Conf. Artificial Intelligence*, Vancouver, 319–325.
- M. Goldberg, D.G. Goodenough, M. Alvo and G. Karam, 1985: A Hierarchical Expert System for Updating Forestry Maps with Landsat Data. *Proceedings of the IEEE*, 73, 1054–1063.
- P. Gong, 1996: Integrated Analysis of Spatial Data from Multiple Sources: Using Evidential Reasoning and Artificial Neural Network Techniques for Geologic Mapping. *Photogrammetric Engineering and Remote Sensing*, 62, 513–523.
- D.G. Goodenough, M. Goldberg, G. Plunkett and J. Zelek, 1987: An Expert System for Remote Sensing. *IEEE Trans Geoscience and Remote Sensing*, GE-25, 349–359.
- H.N. Gross and J.R. Schott, 1998: Application of Spectral Mixture Analysis and Image Fusion Techniques for Image Sharpening. *Remote Sensing of Environment*, 63, 85–94.
- B. Kartikeyan, K.L. Majumder and A.R. Dasgupta, 1995: An Expert System for Land Cover Classification. *IEEE Trans Geoscience and Remote Sensing*, 33, 58–66.

- T. Lee, J.A. Richards and P.H. Swain, 1987: Probabilistic and Evidential Approaches for Multisource Data Analysis. *IEEE Trans Geoscience and Remote Sensing*, GE-25, 283–293.
- W.L. Moon, 1990: Integration of Geophysical and Geological Data Using Evidential Belief Function. *IEEE Trans Geoscience and Remote Sensing*, 28, 711–720.
- N.J. Mulder, H. Middlekoop and J. Miltenberg, 1988: Progress in Knowledge Engineering for Image Classification. 16th Congress of the International Society for Photogrammetry and Remote Sensing, 27 (111), 395–405.
- M. Nagao and T. Matsuyama, 1980: A Structural Analysis of Complex Aerial Photographs. Plenum, New York.
- B. Nicholin and R. Gabler, 1987: A Knowledge-Based System for the Analysis of Aerial Images. *IEEE Trans Geoscience and Remote Sensing*, GE-25, 317–328.
- D. Nute, 1988: Defeasible Reasoning: A Philosophical Analysis in Prolog, in Aspects of Artificial Intelligence. J.H. Fetzer (Ed.) Dordrecht, Kluwer Academic Publishers.
- D.R. Peddle, 1995 a: MERCURY $\oplus$ : An Evidential Reasoning Image Classifier. *Computers and Geosciences*, 21, 1163–1176.
- D.R. Peddle, 1995 b: Knowledge Formulation for Supervised Evidential Classification. *Photogrammetric Engineering and Remote Sensing*, 61, 409–417.
- J.L. Pollack, 1974: Knowledge and Justification. N.J. Princeton University Press.
- J.A. Richards, D.A. Landgrebe and P.H. Swain, 1982: A means of utilizing ancillary information in multispectral classification. *Remote Sensing of Environment*, 12, 463–477.
- A.H. Schistad Solberg, A.K. Jain and T. Taxt, 1994: Multisource Classification of Remotely Sensed Data: Fusion of Landsat TM and SAR Images. *IEEE Trans Geoscience and Remote Sensing*, 32, 768–778.
- R.A. Schowengerdt, 1989: A General Purpose Expert System for Image Processing. *Photogrammetric Engineering and Remote Sensing*, 55, 1277–1284.
- P.S. Sell, 1985: Expert Systems – a Practical Introduction. Maximilian, Southampton.
- G. Shafer, 1976: A Mathematical Theory of Evidence. NJ, Princeton UP.
- B. Solaiman, R.K. Koffi, M-C Mouchot and A. Hillion, 1998: An Information Fusion Method for Multispectral Image Classification Postprocessing. *IEEE Trans Geoscience and Remote Sensing*, 36, 395–406.
- A. Srinivasan, 1990: An Artificial Intelligence Approach to the Analysis of Multiple Information Sources in Remote Sensing. PhD Thesis, The University of New South Wales, Kensington.
- A. Srinivasan and J.A. Richards, 1993: Analysis of GIS Spatial Data Using Knowledge-Based Methods. *Int. J. Geographic Information Systems*, 7, 479–500.
- A.H. Strahler, 1980: The Use of Prior Probabilities in Maximum Likelihood Classification of Remotely Sensed Data. *Remote Sensing of Environment*, 10, 135–163.
- J. Ton, J. Stickten and A.K. Jain, 1991: Knowledge-Based Segmentation of Landsat Images. *IEEE Trans Geoscience and Remote Sensing*, 29, 222–232.
- F. Van Der Meer, 1997: What Does Multisensor Image Fusion Add in Terms of Information Content for Visual Interpretation? *Int. J. Remote Sensing*, 18, 445–452.
- S.W. Wharton, 1987: A Spectral Knowledge Based Approach for Urban Land Cover Discrimination. *IEEE Trans Geoscience and Remote Sensing*, 25, 272–282.

## Problems

**12.1** Compare the attributes of a knowledge-based approach to image interpretation with the more usual approach which uses standard statistical algorithms, such as the maximum

likelihood rule. You should comment on both the training/knowledge acquisition phase and the labelling phase.

**12.2** Write a set of production rules that might be used to smooth a thematic map. The rules are to be applied to the central labelled pixel in a  $3 \times 3$  window. Assume the map has 5 possible classes and that map segments with as few as 4 pixels are acceptable to the ultimate user.

**12.3** Develop a set of production rules that might be applied to Landsat TM imagery to create a thematic map with five classes: vegetation, deep clear water, shallow or muddy water, dry soil and wet soil. To do this you may need to refer to a source of information on the spectral reflectance behaviour of these cover types in the ranges of the TM bands.

**12.4** A rule-based analysis system is a very effective way of handling multi-resolution image data. For example, rules could be applied first to the pixels of the low resolution data to see whether there is a strong endorsement for any of the available labels. If so then the high spatial resolution data source need not be consulted, and data processing time is saved. If, however, the rule-based system can only give weak support to any of the available labels on the basis of the low resolution data, then it could consult the high resolution source to see whether the smaller pixels can be labelled at that level with certainty. This could be the case in an urban region where some low resolution pixels (at say MSS resolution) may be difficult to classify because they are a mixture of vegetation and concrete. The resolution of SPOT HRV may be able to resolve those classes. In some other urban areas, which might be large vegetated regions such as golf courses, the MSS data is quite adequate. Using the strategy of Sect. 12.4, based on justifiers and endorsements, develop a set of rules for such a multi-resolution problem. Your approach should not go beyond the MSS level of resolution if a pixel has a definite or likely endorsement.

This application has been developed fully by Srinivasan (1990).

**12.5** Consider how the perceived quality of data might be taken into account in a qualitative reasoning system. In the justification and endorsement approach, endorsements made on the basis of poor quality data, for example, may lead to the down-grading of an endorsement or justification.

# 13

## Interpretation of Hyperspectral Image Data

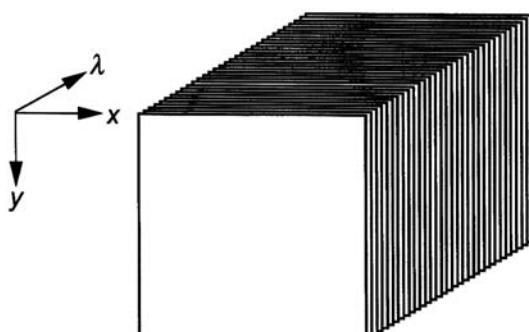
### 13.1

#### Data Characteristics

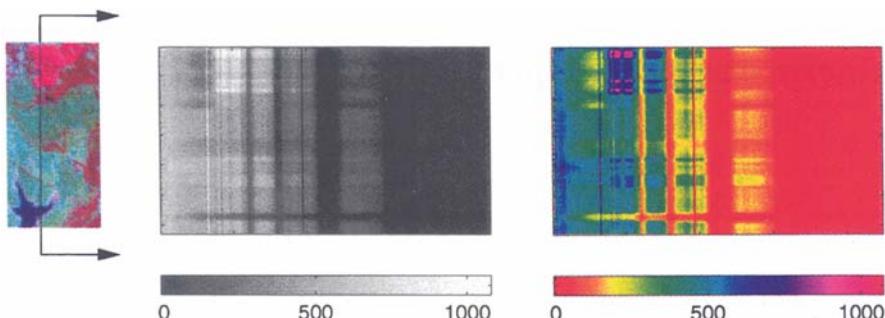
The data produced by the imaging spectrometers of Appendix A is different from that of multispectral instruments owing to the enormous number of wavebands recorded – leading to the term *hyperspectral*. For a given geographical area imaged, the data produced can be viewed as a cube, as shown in Fig. 13.1, having two dimensions that represent spatial position and one that represents wavelength.

When displaying multispectral data, such as that from Landsat, both spatial dimensions are generally used, with three of the spectral bands written to the red, green and blue colour elements of the display device, as described in Fig. 3.1. Sometimes, careful band selection is required in this process to ensure the most informative display, while on other occasions multispectral transformations, such as principal components, are used to enhance the richness of the displayed data.

With hyperspectral data there are both challenges and opportunities presented in creating data displays. First, choosing the most appropriate three channels to use is not straightforward and, in any case, would invariably lead to substantial loss of the spectral benefits offered by this form of data gathering. Nevertheless, unless spectral transformations are employed, a set of three bands comparable to those used with multispectral imagery are often adopted (near IR, red, green) for simple display of



**Fig. 13.1.** Hyperspectral “cube” of image data such as recorded by an imaging spectrometer



**Fig. 13.2.** Line profile display created from hyperspectral data. **a** Transect through portion of a hyperspectral image. **b** Greyscale display of spectral band (horizontally) versus position in the image (vertically). **c** Coloured version of **b**

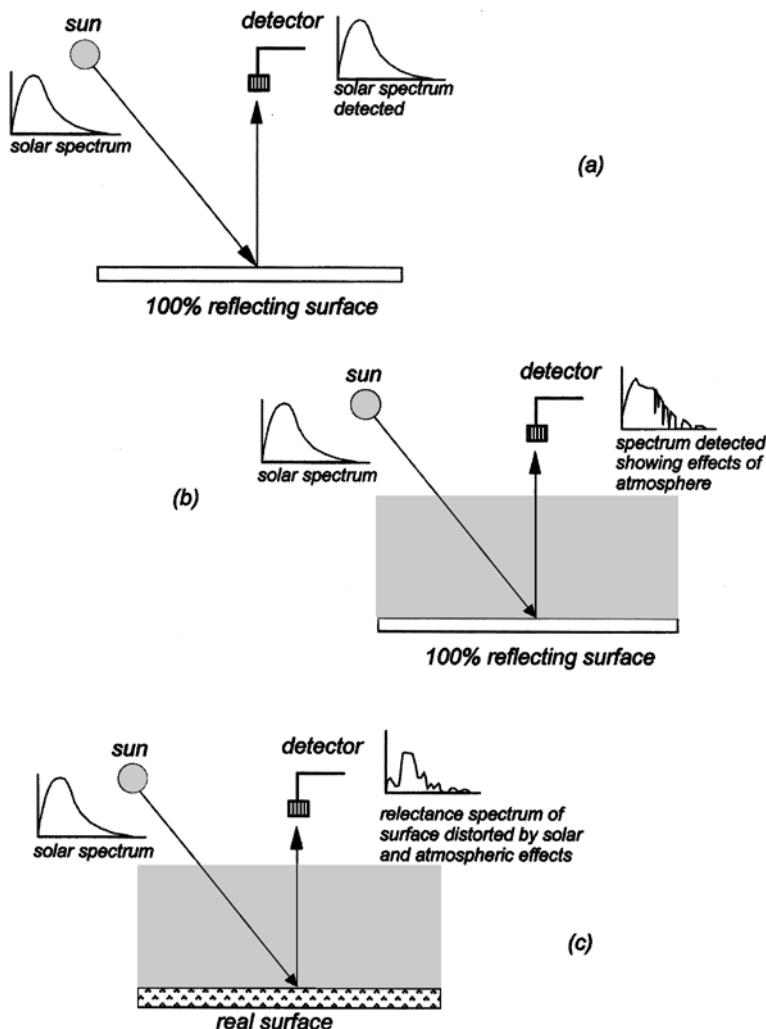
the data. Secondly, because of the large number of bands available, a two dimensional display using one geographical dimension and the spectral dimension can be created as shown in Fig. 13.2. Such a representation allows changes in spectral profiles with position (either along track or across track) to be observed. Usually the greyscale is mapped to colour to enhance the interpretability of the displayed data.

To understand much of what is to follow it is useful to envisage how recorded hyperspectral data is affected by the presence of the atmosphere and the nature of the solar spectrum.

Imagine the region being imaged has a uniform 100% spectral response – in other words it will reflect all of the incident sunlight over all wavelengths, as depicted in Fig. 13.3a; also assume that there is no atmosphere above the surface. A detector capable of taking many spectral samples (say 200 or so) will then essentially record the solar spectrum as shown. If the spectral resolution of the detector were sufficiently fine then the recorded solar spectrum would include the Fraunhofer absorption lines, resulting from the gases in the solar atmosphere (Slater, 1980).

Now suppose there is a normal terrestrial atmosphere in the path between the sun, the surface and the detector. The spectrum recorded will be modified by the extent to which the atmosphere selectively absorbs the radiation. There are well known absorption features caused by the presence of oxygen and water vapour in the atmosphere and these appear in the recorded data as depicted in Fig. 13.3b. Also, the atmosphere scatters the solar radiation leading to the sky irradiance and path radiance terms of Fig. 2.1. So for a start, if we wished to determine the (uniform) spectrum of the ideally reflecting surface, the atmospheric absorption features need to be removed, as does the shape of the solar spectrum and the effect of atmospheric scattering.

Figure 13.3c suggests how the reflectance spectrum of a *real* surface might appear before compensation for solar and atmospheric effects. The spectrum recorded is a combination of the actual spectrum of the real surface, modulated by the effects of the solar curve and the atmosphere. Section 13.3 addresses a range of techniques used for removing those effects.



**Fig. 13.3.** Formation of the reflectance spectrum of a given surface, and the biasing effects of the solar spectral irradiance, atmospheric absorption and scattering

## 13.2 The Challenge to Interpretation

Recall from Chap. 3 that there are essentially two classes of analytical technique used with multispectral data – photointerpretation and machine analysis (classification). The former depends upon the use of image enhancement procedures for improving the visual interpretability of image data whereas the latter is based usually on statistical or other forms of numerical algorithms for labelling individual pixels.

When the data has hundreds of spectral bands traditional image processing and data handling techniques face difficulties. On the other hand, enough information is readily available in the data to allow analysis based on a knowledge of spectroscopic principles, as discussed in Sect. 13.4.1 following.

It is important to understand the limitations placed on the more traditional analytical approaches since those methods still find application with hyperspectral data, not the least reason for which is the substantial investment in image processing software. In the following the features which distinguish hyperspectral from multispectral data are highlighted as a precursor to a discussion on the methods of analysis that can be used with hyperspectral data, either modified or in original form. These differences include data volume, redundancy and dimensionality.

### 13.2.1 Data Volume

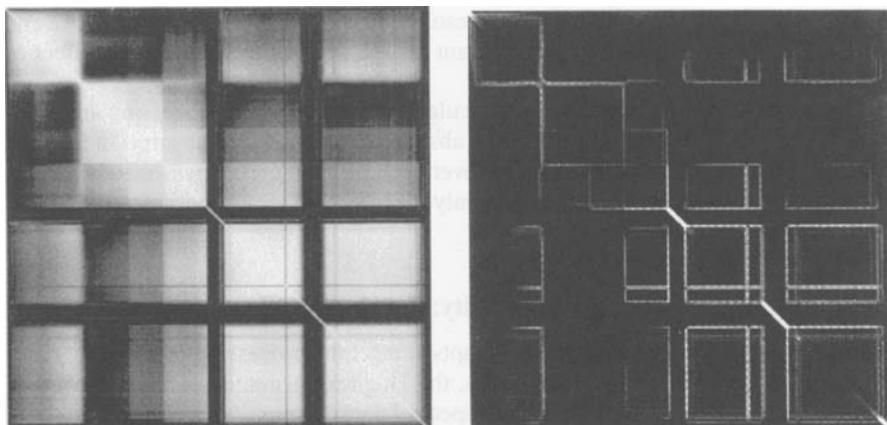
Although data volume strictly does not pose any major data processing challenges with contemporary computing systems it is nevertheless useful to examine the relative magnitudes of data for say Landsat Thematic Mapper multispectral imagery and AVIRIS hyperspectral data.

Clearly, the major differences to note between the two is the number of wavebands (7 versus 224) and the radiometric quantisations used (8 versus 10 bits per pixel per band). Ignoring differences in spatial resolution, the relative data volumes, per pixel, are  $7 \times 8 : 224 \times 10$  – i.e. 56 : 2240. Per pixel there are 40 times as many bits therefore for AVIRIS as for TM data. Consequently, storage and transmission of hyperspectral data are issues for consideration; suitable data compression techniques are discussed in Sect. 13.7.

### 13.2.2 Redundancy

With 40 times as much data per pixel one is led to question whether 40 times as much information can be obtained about the ground cover types being imaged. Generally, of course, that is not the case – much of the additional data does not add to the inherent information content for a particular application even though it often helps in discovering that information. In other words it contains redundancies.

Much of the data we deal with in everyday life is highly redundant. Take the English language as an example. If we remove certain letters from a word we can often still understand what word is intended. For example *rmte sesng* would be recognised by most people who read this book as *remote sensing* because there are sufficient redundant letters that losing some is not critical to understanding. The same is true with remote sensing data, especially that recorded by hyperspectral sensors – there is often substantial overlap of information content over the bands of data recorded for a given pixel. In such cases not all of the data is needed to characterise a pixel properly, although redundant data may be different for different applications.



**Fig. 13.4.** **a** The correlation matrix for 196 wavebands<sup>1</sup> covering 400 nm to 2400 nm for the AVIRIS Jasper Ridge image (white represents correlations of 1 or  $-1$ , while black indicates a correlation of 0). **b** The result of edge detecting the correlation matrix

In remote sensing data redundancy can take two forms: spatial and spectral. Exploiting spatial redundancy is behind the spatial context methods of Sect. 8.8. Spectral redundancy means that the information content of one band can be fully or partly predicted from the other bands in the data. An example of this is seen in Fig. 6.2b.

An interesting way to view spectral redundancy is to form the correlation matrix for an image (or portion of an image) of interest; the correlation matrix can be derived from the covariance matrix using (6.3). High correlations between band pairs indicate high degrees of redundancy. Because there are so many bands with hyperspectral data it is not practical to list all the correlations numerically, such as is done in Sect. 6.1.1. Instead, it is better to display the inherent correlations (redundancies) pictorially as shown in Fig. 13.4a, where a grey scale is used to represent levels of correlation. This representation is often used with hyperspectral data and is a useful tool for identifying correlations among bands when applying traditional processing tools as will be seen later. An interesting by-product of representing the correlation (or covariance) matrix in this form is that image processing procedures can be applied to it. For example its block structure can be emphasised by using a simple edge detection filter to give the result shown in Fig. 13.4b.

Means for removing inherent redundancy are often not readily apparent, although techniques such as the principal components transformation assist in the task since decorrelation followed by discarding low variance components amounts to redundancy-reduction.

---

<sup>1</sup> Overlapping bands result from the use of four individual spectrometers in the AVIRIS instrument; these and the significant water absorption bands and bands which have very small means ( $< 2$ ) have been deleted from the original 224 bands, leaving 196 bands for image processing.

### 13.2.3

#### The Need for Calibration

The high spectral resolution of hyperspectral data sets means that fine atmospheric absorption features will be detected and displayed as discussed in Sect. 13.1. In order that they not be confused with absorption features of the ground cover type being imaged it is important to account for them and “remove” them from the data.

Moreover, because the high spectral resolution suggests that recorded spectra can be interpreted scientifically it is important also to remove the modulating effect of the solar spectrum.

Neither of those effects has been particularly important in the processing and analysis of multispectral data because of the absence of well defined absorption features and the use of average solar irradiance over each of the recorded wavebands as suggested in (2.1). With multispectral data only the effects of atmospheric scattering and transmittance are corrected.

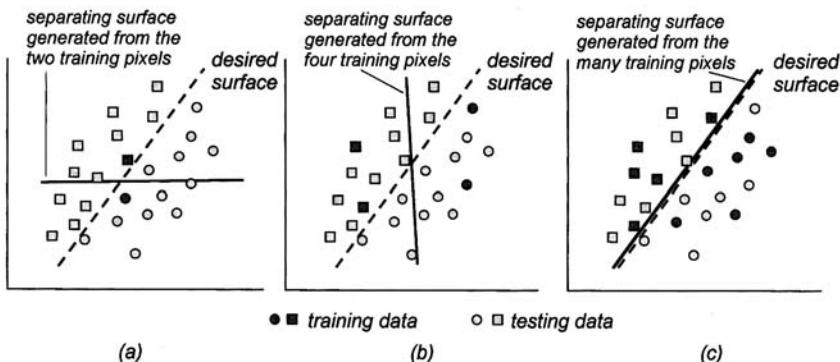
### 13.2.4

#### The Problem of Dimensionality: The Hughes Phenomenon

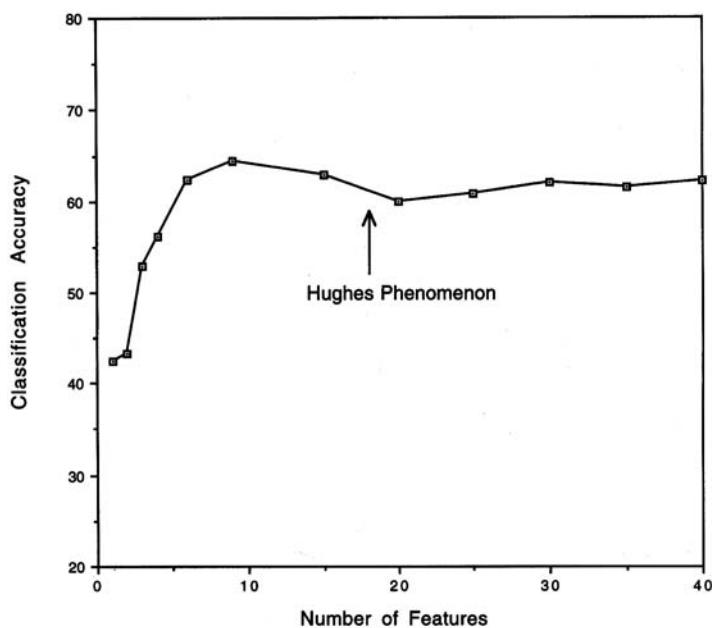
While recognised since the earliest attempts at machine processing of remotely sensed image data (Swain and Davis, 1978), the Hughes phenomenon had not been of major concern until the advent of hyperspectral data.

Briefly, a minimum ratio of the number of training pixels to number of spectral bands is needed to ensure reliable estimates of class statistics are obtained when training supervised classifiers; as the dimensionality of the data set increases the minimum number of training pixels per class must be increased to preserve the accuracy of the statistical estimates. Thus, adding more spectral bands, as in the case of AVIRIS, MODIS and Hyperion, is not helpful unless more training pixels per class are available. This turns out to be one of the major limitations in attempting to apply traditional image classification procedures to hyperspectral data. A simple example, based on determining a reliable linear separating surface, can be used to illustrate the problem. Figure 13.5 shows three different training sets of data for the same two dimensional (band) data set. The first (Fig. 13.5a) has only one pixel per class. As seen, while a separating surface can be found it may not be accurate. Having two training pixels per class as in the case of Fig. 13.5b provides a better estimate of the separating surfaces, but it is not until we have many pixels per class, when compared to the number of channels in the data, that we will obtain good estimates of the parameters of the supervised classifier (Fig. 13.5c).

This is simply another way of looking at the material of Sect. 8.2.6. However, rather than increase the number of training pixels for a given number of bands, consider now the case of increasing the number of bands for a set number of training pixels; the same problem is observed as illustrated in Fig. 13.6. We note that the performance of the classifier is compromised by the poor estimates of the training statistics beyond about ten features.



**Fig. 13.5.** Illustration of the importance of enough training samples per class to ensure reliable estimation of a separating surface. When too few pixels are used (**a**) good separation of the training data is possible but the classifier performs poorly on the testing data. Large numbers of (randomly positioned) training pixels generate a surface that also performs well for testing data (**c**)



**Fig. 13.6.** The Hughes phenomenon, demonstrating logs of classifier performance (on testing data) with increasing data dimensionality. This graph is the result of a four category classification; the features indicated are the best sets of those sizes, selected using the Bhattacharyya separability measure.

## 13.3

### Data Calibration Techniques

#### 13.3.1

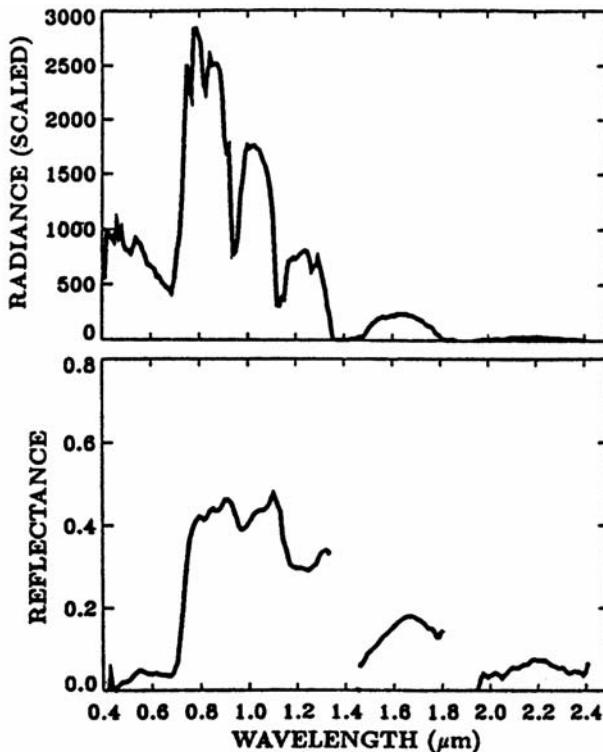
##### Detailed Radiometric Correction

As discussed in Sects. 2.1.1 and 13.2.3, the upwelling radiance measured by a sensor results from incident solar energy scattered and reflected from the atmosphere and earth surface. Detailed radiometric correction to obtain surface reflectance for hyperspectral data follows similar procedures as for the examples given in Sect. 2.2.1. However, since hyperspectral data covers the whole spectral range from 0.4 to 2.4  $\mu\text{m}$ , including water absorption features, and has high spectral resolution, a more systematic process is generally required, consisting of three possible steps:

- Compensation for the shape of the solar spectrum. The measured radiances are divided by solar irradiances above the atmosphere to obtain the *apparent* reflectances of the surface.
- Compensation for atmospheric gaseous transmittances and molecular and aerosol scattering. Simulating these atmospheric effects allows the *apparent* reflectances to be converted to *scaled* surface reflectances.
- Scaled surface reflectances are converted to *real* surface reflectances after consideration of any topographic effects. If topographic data is not available, real reflectance is assumed to be identical to scaled reflectance under the assumption that the surfaces of interest are Lambertian.

Procedures for solar curve and atmospheric modelling are incorporated in a number of models (Gao et al., 1993), including Lowtran 7 (Low Resolution Atmospheric Radiance and Transmittance), 5S Code (Simulation of the Satellite Signal in the Solar Spectrum) and Modtran 3 (The Moderate Resolution Atmospheric Radiance and Transmittance Model – see Anderson et al., 1995).

ATREM (Atmosphere REMoval Program, Gao et al., 1992), which is built upon 5S code, overcomes a difficulty with the other approaches in removing water vapour absorption features in AVIRIS data; water vapour effects vary from pixel to pixel and from time to time. In ATREM the amount of water vapour on a pixel-by-pixel basis is derived from AVIRIS data itself, particularly from the 0.94  $\mu\text{m}$  and 1.14  $\mu\text{m}$  water vapour features. A technique referred to as three-channel ratioing is developed for this purpose (Gao et al., 1993). Figure 13.7 shows an example of a corrected spectrum against the original measurements.



**Fig. 13.7.** **a** Raw AVIRIS vegetation spectrum and **b** its correction based on ATREM.  
(Reprinted from Gao et al., 1993 with permission from Elsevier Science)

### 13.3.2 Data Normalisation

When detailed radiometric correction is not feasible (for example, because the necessary ancillary information is unavailable) normalisation is an alternative which makes the corrected data independent of multiplicative noise, such as topographic and solar spectrum effects. This can be performed using *Log Residuals* (Green and Craig, 1985), based on the relationship between radiance (raw data) and reflectance:

$$x_{i,n} = T_i R_{i,n} I_n, \quad i = 1, \dots, K; n = 1, \dots, N$$

where  $x_{i,n}$  is radiance for pixel  $i$  in waveband  $n$ .  $T_i$  is the topographic effect, which is assumed constant for all wavelengths.  $R_{i,n}$  is the real reflectance for pixel  $i$  in waveband  $n$ .  $I_n$  is the (unknown) illumination factor, which is assumed independent of pixel.  $K$  and  $N$  are the total number of the pixels in the image and the total number of bands, respectively.

There are two steps which remove the topographic and illumination effects respectively.  $x_{i,n}$  can be made independent of  $T_i$  and  $I_n$  by dividing  $x_{i,n}$  by its geometric mean over all bands and then its geometric mean over all pixels. The result is not

identical to reflectance but is independent of the multiplicative illumination and topographic effects present in the raw data. The procedure is carried out logarithmically so that the geometric means are replaced by arithmetic means and the final result obtained for the normalised data is

$$\begin{aligned}\log z_{i,n} &= \log x_{i,n} - \log m_n - \log m_i \\ &= \log x_{i,n} - \frac{1}{N} \sum_{n=1}^N \log x_{i,n} - \frac{1}{K} \sum_{i=1}^K \log x_{i,n}\end{aligned}$$

### 13.3.3 Approximate Radiometric Correction

As with multispectral data approximate correction is acceptable for some applications. One approach is the Empirical Line procedure (Roberts et al., 1985). Two spectrally uniform targets in the site of interest, one dark and one bright, are selected; their actual reflectances are then determined by field or laboratory measurements. The radiance spectra for each target are extracted from the image and then mapped to the actual reflectances using linear regression techniques. The gain and offset so-derived for each band are then applied to *all* pixels in the image to calculate their reflectances.

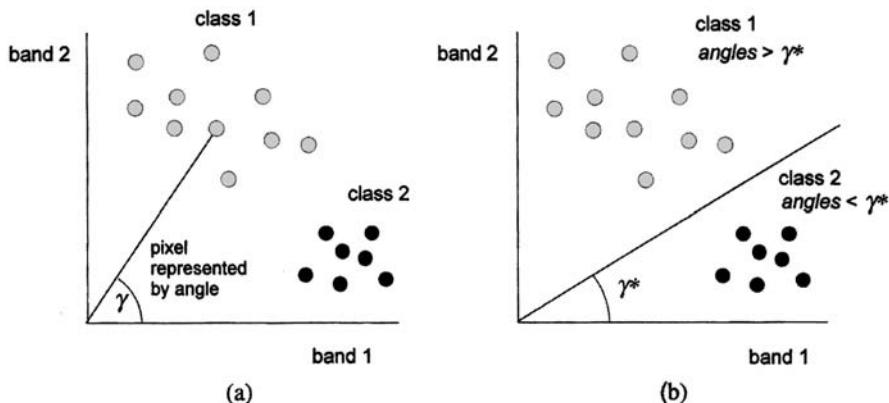
While the computational load is manageable with this method, field or laboratory data may not be available. The Flat Field method (Roberts et al., 1986), an approximate correction technique that relies purely on the image data itself, is then an alternative. This depends on locating a large, spectrally uniform area in an image (such as sand or clouds) and finding its average radiance spectrum. It is assumed that the shape and the absorption features presented in this spectrum are caused by solar and atmospheric effects. The reflectance of each pixel is then obtained by dividing the average radiance spectrum into the image spectrum of the pixel.

## 13.4 Interpretation Using Spectral Information

### 13.4.1 Spectral Angle Mapping

As will be seen in later sections, pixel labelling techniques in hyperspectral data analysis, based on standard classification procedures, can often fail because of the difficulty in obtaining reliable class definitions with data of such high dimensionality. One means for coping with the problem is to reduce the dimensionality by some means. A candidate approach is to ignore the magnitudes of the pixel vectors in hyperspectral space and attempt classification instead using just their angular orientations as their sole describing characteristic.

In  $N$  dimensional multi-(hyper-)spectral space a pixel vector  $\mathbf{x}$  has both magnitude (length) and an angle measured with respect to the axes that define the coordinate



**Fig. 13.8.** **a** Representing pixels by their angles from the band axes. **b** Segmenting the multi-spectral space by angle

system of the space (see Appendix D). In the spectral angle mapper (SAM) technique for identifying pixel spectra only the angular information is used. Figure 13.8a shows a two dimensional (ie. two band) example where spectra are characterised entirely by their angles from the horizontal axis. The spectra can be distinguished from each other provided the angles are sufficiently different. Using this concept, angular decision boundaries can be set up (from library information or training data) that segment the space as shown in Fig. 13.8b. Spectra are then labelled according to the sector within which they fall.

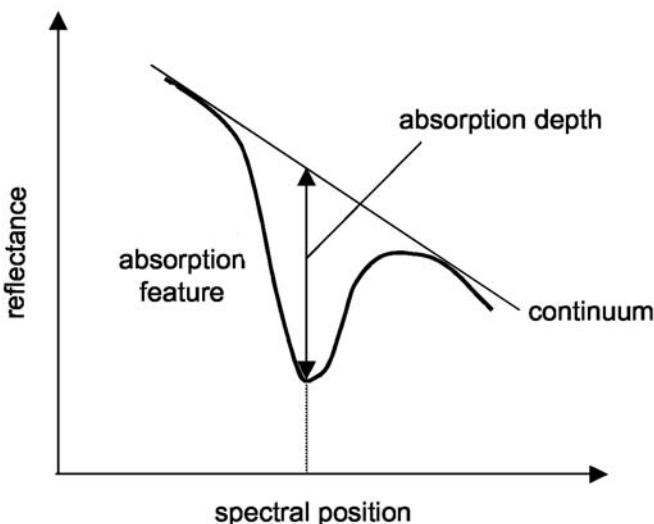
Clearly the SAM technique will fail if the vector magnitude is important in providing discriminating information, which it will in many instances. However, if the pixel spectra from the different classes are well distributed in the space there is a high likelihood that angular information alone will provide good separation. The technique functions well in the face of scaling noise. Details for implementing SAM will be found in Kruse et al. (1993).

### 13.4.2

## Using Expert Spectral Knowledge and Library Searching

Having a well-defined spectrum means that a scientific approach to interpretation can in principle be carried out, much as a sample is identified using spectroscopy in the laboratory through a knowledge of spectral features.

Absorption features (seen as localised dips) are often observed in the reflectance spectra of specific minerals provided sufficient spectral resolution is available. It is those absorption features that provide the information needed for identification. They are sometimes referred to therefore as “diagnostically significant features”. Characterisation and thus automatic detection of such absorption features, when they occur, is of particular interest in hyperspectral image recognition.



**Fig. 13.9.** Illustration of the importance of defining the continuum in a spectrum before measuring the properties of diagnostically significant features

Absorption features can be characterised by their locations (bands), relative depths and widths (full width at half the maximum depth), and used in pixel identification.

To make that possible it is important to separate the absorption features from the background continuum of the spectrum that results from light transmission and scattering, as against the absorption features themselves that are due to photon interaction with the atomic structure of the chemicals present in the material being observed. The importance of continuum removal can be seen in Fig. 13.9. Often the background will not be “horizontal”, so definition of the depth of the feature can then be ambiguous. If the continuum in the vicinity of the feature is defined by a line of best fit between those portions of the spectrum either side of the feature then a reasonably consistent measure of band depth can be established.

Usually, a complete spectrum is divided into several spectral regions (often under the guidance of an expert) and absorption features are detected in each of the regions. An unknown pixel is then labelled as belonging to a given class if the properties of its diagnostically significant absorption features match those of the spectrum for that class held in a spectral feature library.

A complication that can arise with library searching in general, and with seeking to match absorption features in particular, is that mixtures are often encountered, and some materials have very similar spectral features. An excellent treatment of the complexities that arise, and how they can be handled, is given in Clark et al (2003).

### 13.4.3

#### Library Searching by Spectral Coding

Because the pixel spectrum is so well specified and can be corrected for atmospheric and solar distortions, spectral comparison is possible – either with previously recorded data or with laboratory spectra – for pixel identification. The reference spectra are usually stored in a spectral library.

It is clear that the searching and matching processes must be efficient in such a procedure. Full spectral matching using original radiometric data is not practical. However, given the degree of redundancy spectrally and radiometrically that one would anticipate with the data recorded by an imaging spectrometer, coding techniques can be employed to represent a pixel spectrum in a simple and effective manner so that fast library searching and matching can be achieved.

##### 13.4.3.1

###### Binary Spectral Codes

A simple binary code for a reflectance spectrum can be formed according to

$$h(n) = \begin{cases} 0 & \text{if } x(n) \leq T \\ 1 & \text{otherwise} \end{cases} \quad n = 1, \dots, N$$

where  $x(n)$  is the brightness value of a pixel in the  $n^{th}$  band,  $T$  is a user specified threshold for forming the binary code, and  $h(n)$  is the resulting binary code symbol for the pixel in the  $n^{th}$  spectral band. Usually  $T$  is chosen as the average brightness value of the spectrum. Figure 13.10 demonstrates a typical spectrum encoded in this manner. Instead of using the average brightness of the complete spectrum as a threshold, the local average over the adjacent channels could be employed.

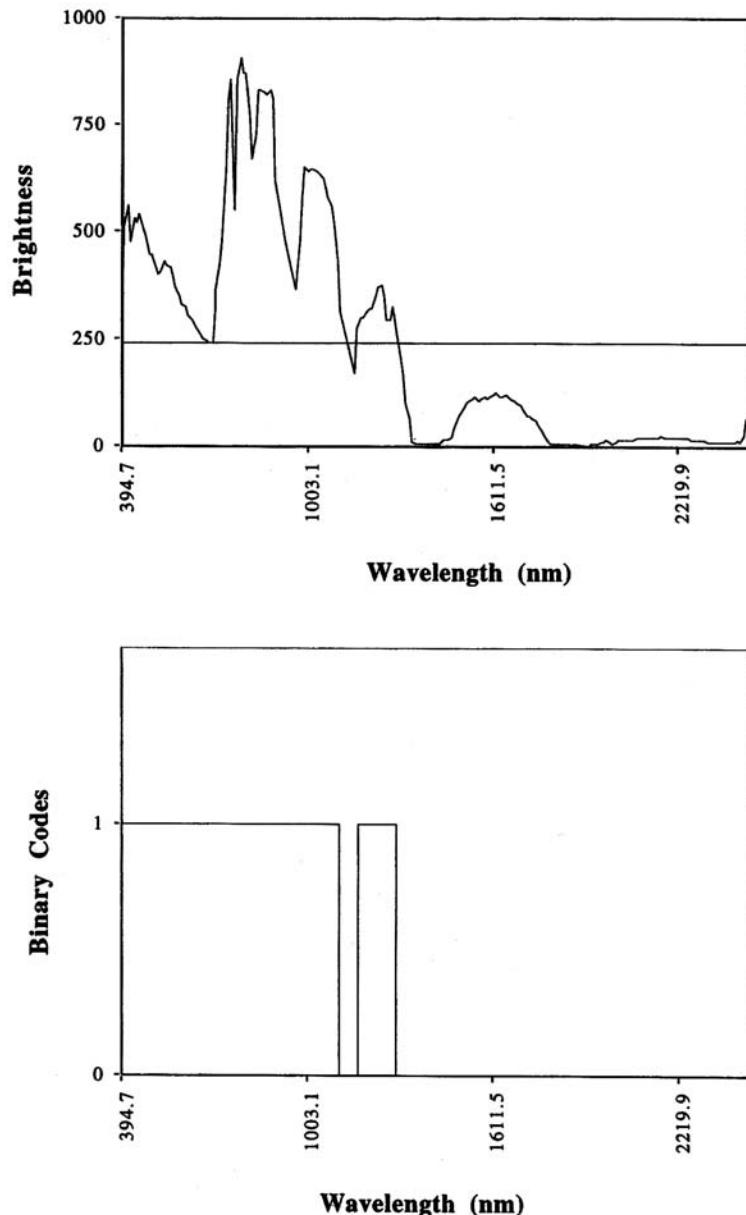
Such a simple binary code will not always provide reasonable separability between the spectra in a library, nor will it guarantee that a measured spectrum will match with either only one or a small number of library spectra. Consequently, more sophisticated codes may need to be adopted. For example, more than one threshold could be used. With three thresholds a two binary digit code for the brightness of a pixel will be created:

$$\begin{aligned} h(n) = & 00 & \text{if } x(n) \leq T_1 \\ & 01 & \text{if } T_1 < x(n) \leq T_2 \\ & 11^2 & \text{if } T_2 < x(n) \leq T_3 \\ & 10 & \text{if } T_3 < x(n). \end{aligned}$$

The mean brightness over the spectrum can be one threshold; the other two are chosen above and below this value.

---

<sup>2</sup> The third level code word is chosen as 11 rather than 10 so that there is only one binary digit difference between levels.



**Fig. 13.10.** Formation of a simple binary code for an AVIRIS spectrum

Spectral slope can also be used as part of a code. One binary representation of the local slope,  $s(n)$ , at each waveband is:

$$\begin{aligned} s(n) = 0 & \quad \text{if } (x(n+1) - x(n-1)) \leq 0 \\ 1 & \quad \text{otherwise} \end{aligned} \quad n = 1, \dots, N$$

Another variation is to develop separate codewords for different regions of the spectrum. The resulting codes accommodate spectral coarse structure better. Uniformly spaced regions could be used or perhaps those regions of the spectrum suspected as being most significant in differentiating cover types could be adopted. The latter is based upon the knowledge that in different wavelength ranges the reflectance spectrum is dominated by different physical characteristics of the surface being imaged. Figure 13.11 shows an example of coding on selected bands with 3 thresholds.

### 13.4.3.2 Matching Algorithms

Comparison of binary coded spectra can be made by measuring the Hamming distance between them, defined as

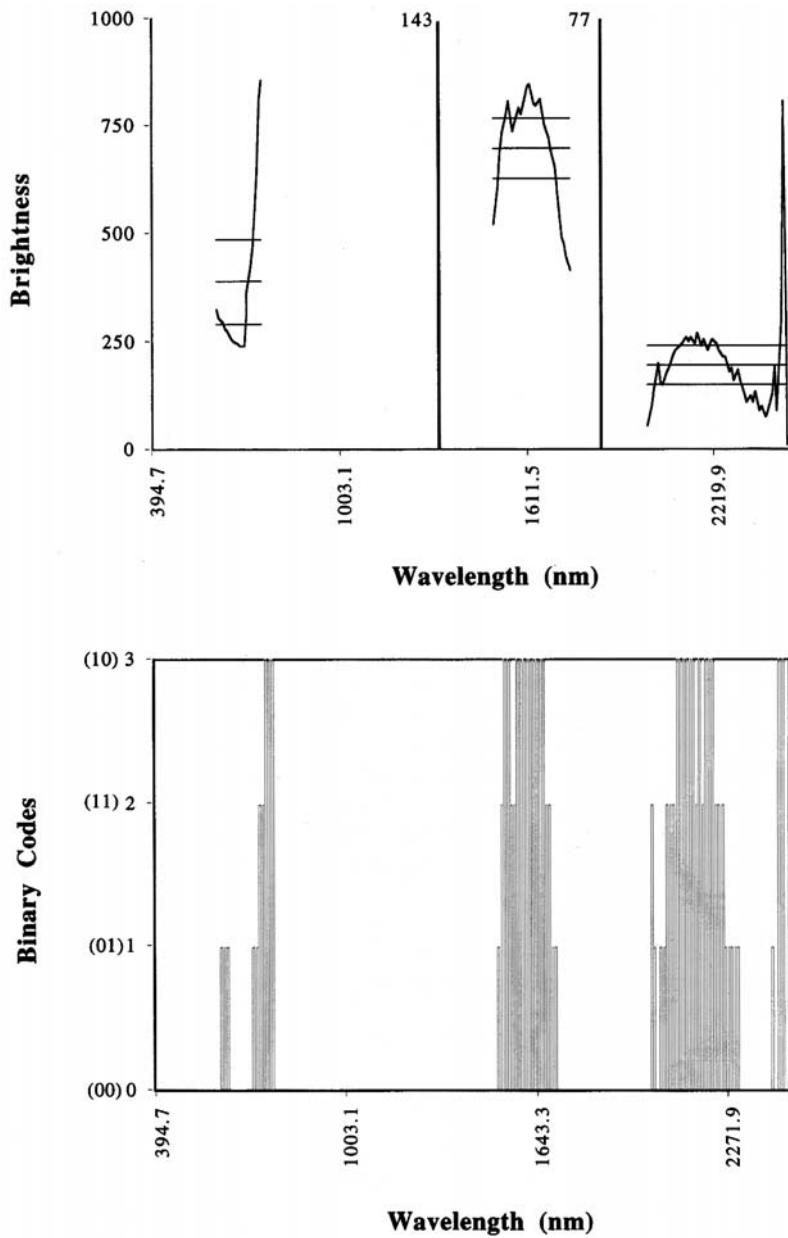
$$D_H(h_i, h_j) = \sum_{l=1}^L (h_i(l) \oplus h_j(l))$$

where  $h_i$  and  $h_j$  are two spectral codewords of length (i.e. number of bits)  $L$ . For simple thresholding,  $L = N$ , the number of bands.  $L = 2N$  if slope coding is also used or three thresholds are employed.  $\oplus$  denotes the exclusive OR operator. It is applied on a bit-by-bit basis for a pair of binary code words and records a difference as ‘1’ and no difference as ‘0’. For example, the exclusive OR of two spectral codewords 01110011 and 00101011 becomes 01011000. Hamming distance is then calculated by summing the number of times the binary digits are different. In this example, the Hamming distance is 3. If the distance is within a user-specified threshold, the two pixels are identified as belonging to the same class. When one, say  $h_i(n)$ , is a class signature code, the comparison leads to labelling for pixel  $j$ .

## 13.5 Hyperspectral Interpretation by Statistical Methods

### 13.5.1 Limitations of Traditional Thematic Mapping Procedures

Traditional supervised and unsupervised classification techniques will require very long processing times for hyperspectral data because of the dependence on the number of wavebands (see Sects. 8.5 and 9.3.5). A more serious problem, however, is the need to estimate class signatures – i.e. the mean vector and covariance matrix – when using algorithms, such as maximum likelihood, based on second order statistics. The



**Fig. 13.11.** Formation of a binary code using three thresholds, chosen differently in different regions of the spectrum

difficulty lies in the small number of available training pixels per class compared with the number of wavebands used, and is related directly to the Hughes phenomenon of Sect. 13.2.4. If too few training samples are used then the class model may be very accurate for the training data and classification accuracy on training data can be very high. However, classification accuracy on testing data will be poor. In this case, the classifier is overtrained and the statistics estimated are unreliable. This difficulty is analogous to that of curve fitting illustrated in Sect. 2.4.1.4. To avoid the problem of unreliable class statistics and thus poor classifier performance the number of training pixels per class should be at least ten times the dimensionality of the data, with desirably 100 times as discussed in Sect. 8.2.6.

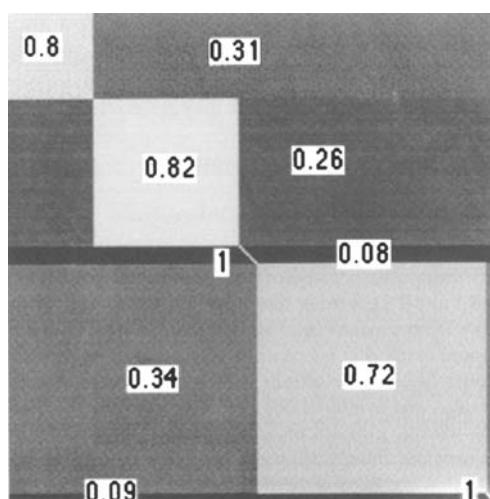
The following sections treat some techniques developed for dealing with the small training set problem.

### 13.5.2

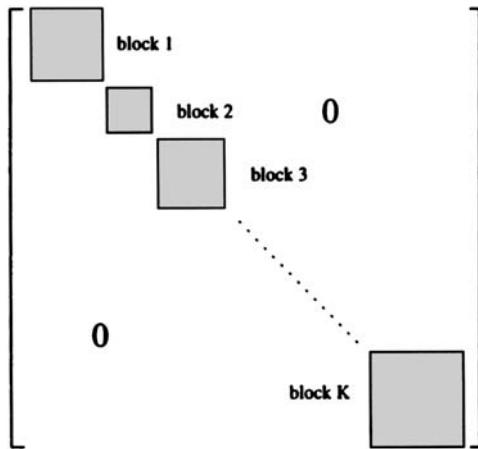
#### Block-based Maximum Likelihood Classification

In general, correlations between neighbouring bands in hyperspectral data sets are higher than for bands further apart and highly correlated bands appear in groups. As a result, the correlation matrix is roughly block diagonal in form as shown in Fig. 13.4, in which a greyscale is used to represent the degree of correlation. Figure 13.12 shows the data of Fig. 13.4a but, for purposes of illustration, with the correlations averaged within identifiable blocks demonstrating the strongly block diagonal form of the correlation and thus the covariance matrix. Those blocks can be identified visually or with the assistance of edge detection on the correlation matrix as shown in Fig. 13.4b.

Now assume that the low off-diagonal correlations are zero. The matrix is then fully block diagonal as depicted in general terms in Fig. 13.13. By assuming that



**Fig. 13.12.** Average correlations within diagonal blocks and within selected off-diagonal segments of Fig. 13.4 illustrating the pseudo block diagonal nature of the matrix



**Fig. 13.13.** Assumed block diagonal form of the correlation and thus covariance matrix

the subgroups of bands within each block are independent of those in other subgroups, maximum likelihood classification can then be applied to each subgroup independently.

Noting that the block diagonal form of the correlation matrix leads to a covariance matrix of the same structure, the discriminant function becomes the sum of the logarithmic discriminant values of the individual groups of wavebands (blocks):

$$g_i(\mathbf{x}) = -\sum_{k=1}^K \{\ln |\Sigma_{ik}| + (\mathbf{x}_k - \mathbf{m}_{ik})^t \Sigma_{ik}^{-1} (\mathbf{x}_k - \mathbf{m}_{ik})\}$$

$$i = 1, \dots, M; \quad k = 1, \dots, K \quad (13.1)$$

In (13.1) the dimensions of  $\mathbf{x}$ ,  $\mathbf{m}_i$ , and  $\Sigma_i$  are reduced to  $n_k$  ( $n_k < N$ ), the size of the  $k^{th}$  subgroup of bands, so that advantage can be taken of the corresponding quadratic reduction in classification time (see Sect. 8.5). Also, the number of training pixels required per class for reliable statistics, determined by the size of the biggest subgroup, is much smaller than when all bands are used.

The sizes of subgroups to use are generally guided by observation of the boundaries of the high correlation blocks along the principal diagonal of a correlation matrix, which will be different for different images.

If training data is limited some relatively high correlations may have to be ignored. However, this approach will still be better than, say, minimum distance classification (often used when training pixels are limited – see Sect. 8.3.1) since at least some correlations are taken into account.

With some data sets, highly correlated blocks of bands will occur away from the diagonal. They can be moved onto the diagonal by reordering the bands before the correlation matrix is computed. Such an operation makes no difference to the information contained in the matrix or to subsequent image analysis operations. However, it does mean that a reconstructed pixel spectrum will have some bands out of order in the sequence of wavelengths.

**Table 13.1.** Re-ordered and original blocks of bands

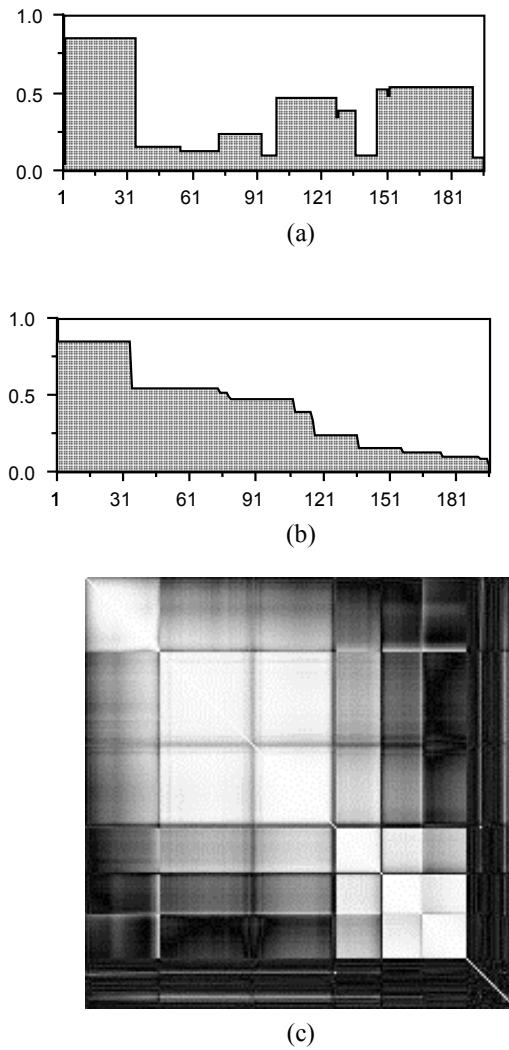
Group Number	Reordered Band Number	Original Band Number
1	1–34	2–35
2	35–38	148–151
3	39–77	153–191
4	78–105	101–128
5	106–113	130–137
6	114	129
7	115	152
8	116–135	74–93
9	136–153	56–73
10	154–173	36–55
11	174	1
12	175–181	94–100
13	182–191	138–147
14	192–196	192–196

A simple and effective means for re-ordering the bands is to consider the first set of rows in the image of the correlation matrix of Fig. 13.4a corresponding to the first highly correlated (diagonal) block of bands. That block covers bands 2–35 in this example. Moving across those 34 rows as a single group, blocks of similar correlation are identifiable (they are correlations of the respective bands with bands 2–35). If we average the correlations in those blocks, the graph of Fig. 13.14a is produced. If we then re-arrange the bands as shown in Fig. 13.14b, by moving the more highly correlated blocks of bands to the left and the less correlated blocks to the right then that has the effect of re-arranging the blocks of bands in the correlation matrix such that the lower correlated blocks are shifted towards the off-diagonal corners and the more highly correlated blocks are moved to the diagonal as shown in Fig. 13.14c.

For interest, Table 13.1 shows how the band blocks for this example have been re-ordered.

## 13.6 Feature Reduction

Given that hyperspectral data is often highly redundant, feature reduction will be an important preprocessing step to image analysis. However, feature reduction itself for hyperspectral data is a time consuming process and feature extraction via linear transform relies, as with classification, on good estimates of class statistics. To solve



**Fig. 13.14.** **a** Average correlations of the blocks of bands evident horizontally in Fig. 13.4a in a strip corresponding to bands 2–35. **b** Blocks of bands re-ordered to rank the average correlations from highest to lowest. **c** Correlation matrix generated with the reordered band positions

in this problem the block-based technique presented in Sect. 13.5.2 can be extended to deal with hyperspectral feature reduction.

### 13.6.1 Feature Selection

Separability measures, such as the JM distance of (10.5) and (10.6), provide metrics of the average distance between two class density functions, and are thus used to find the best subsets of features.

When the complete set of bands is treated as  $K$  independent blocks as discussed in Sect. 13.5.2, the JM distance or other separability measures can be simplified; (10.6) for example becomes

$$B = \sum_{k=1}^K \left\{ \frac{1}{8} (\mathbf{m}_{ik} - \mathbf{m}_{jk})^t \left\{ \frac{\Sigma_{ik} + \Sigma_{jk}}{2} \right\}^{-1} (\mathbf{m}_{ik} - \mathbf{m}_{jk}) \right. \\ \left. + \frac{1}{2} \ln \left\{ \frac{|(\Sigma_{ik} + \Sigma_{jk})/2|}{|\Sigma_{ik}|^{1/2} |\Sigma_{jk}|^{1/2}} \right\} \right\}$$

Thus the Bhattacharyya distance between a class pair is the sum of the distances computed for each block (group of bands).

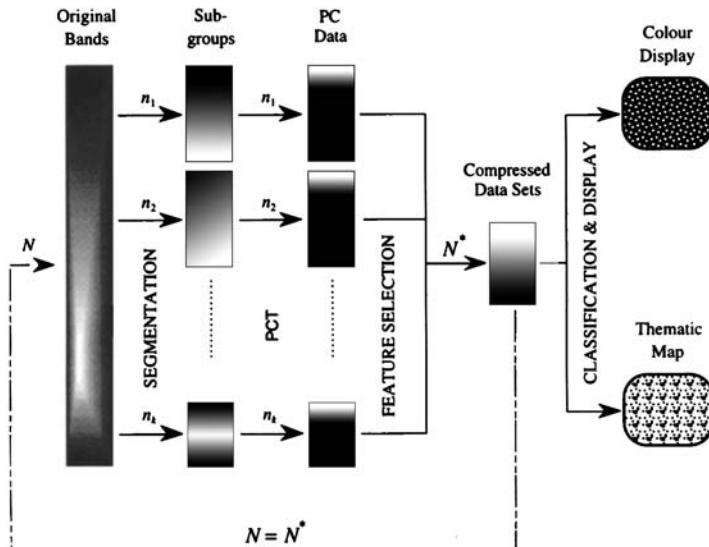
### 13.6.2 Spectral Transformations

The principal components transformation, which uses global statistics to determine the transformation operation, is sometimes used in multispectral data analysis as a tool for feature reduction. The main concern in employing it with hyperspectral data is its high computational load.

Implementing the transformation consists of two tasks: eigenanalysis to generate the transformation matrix  $G$  in (6.4), and pixel by pixel linear transformation. The former requires an insignificant amount of work. However, the latter is a time consuming process which requires  $N \times N$  multiplications and  $N \times (N - 1)$  additions per pixel. Moreover, the process can be biased by high variance bands. For example, the data recorded by AVIRIS is affected in shape by the solar spectrum as shown in Fig. 13.3c. This indicates that a spectral weighting is imposed. As a result, the variances of the spectral bands in the short wavelength region are much higher than the remaining bands if the data is not calibrated. A conventional principal components transform will be dominated, therefore, by the visible and near infrared bands.

When the original bands are highly correlated, the principal components transform works effectively, while for poorly correlated data there may be little change after application of the transform. Recall, for hyperspectral data, high correlations generally occur in blocks. If the conventional principal components transform is modified so that the low correlations between the highly correlated blocks are avoided, the efficiency of the transformation will be improved while the results should be little affected. This leads to the formation of a segmented principal components transformation.

Figure 13.15 shows the process schematically. The complete data set is first partitioned into  $K$  subgroups of highly correlated bands. Denote by  $n_1, n_2, \dots, n_K$  the number of bands in subgroups 1, 2, ...,  $K$ , respectively. The principal components transformation is now conducted separately on each subgroup of data. Feature selection on each of the transformed data sets is carried out by either making use of variance information in each component as is common in multispectral data processing, or by pursuing single band separabilities (see Sect. 13.6.3). The features selected



**Fig. 13.15.** Schematic representation of segmenting the principal components transformation for feature reduction

can be regrouped and transformed again to compress the data further. Generally, the steps can be repeated until the required data reduction ratio is achieved for classification or storage purposes. For colour composite display the most informative three features will be used.

Segmenting the principal components transform in this manner requires  $n_k \times n_k$  multiplications for each subgroup and thus a total of  $\sum_{k=1}^K n_k^2$  multiplications for each pixel vector in contrast to  $N \times N$  multiplications for each pixel vector if transformation over the full set of bands is performed. As an example, 2/3 of the total time is saved when three subgroups of uniform size are used (i.e.,  $K = 3$ , and  $n_1 = n_2 = n_3 = N/3$ ).

So long as all the new transformed components are kept, there is no variance (information) loss by transforming sub-vectors separately. When the new components obtained from each segmented transform are gathered and transformed again, the resulting data variance and covariance are identical to those for the conventional principal components transform.

The segmentation idea can be extended to canonical analysis. The complete set of bands is segmented into  $K$  groups. Then conventional canonical analysis is applied to each group, with up to  $M - 1$  best features selected from each transformed set, where  $M$  is the number of classes. By so doing, class statistics involving the complete set of bands are no longer needed (which otherwise presents the difficulties under limited training pixels discussed in Sect. 13.5.1).

### 13.6.3

#### Feature Selection from Principal Components Transformed Data

For original, untransformed data, feature selection is based on pairwise separability measures such as the Bhattacharyya distance (10.6). If the covariance matrices,  $\Sigma_i$  and  $\Sigma_j$ , are diagonal (following transformation) then (10.6) becomes

$$B = \sum_{n=1}^N \left[ \frac{(m_i(n) - m_j(n))^2}{4(\sigma_i^2(n) + \sigma_j^2(n))} + \frac{1}{2} \ln \frac{(\sigma_i^2(n)/2 + \sigma_j^2(n)/2)}{\sqrt{\sigma_i^2(n)\sigma_j^2(n)}} \right]$$

where  $m_i(n)$ ,  $\sigma_i^2(n)$  represent, respectively, the mean and variance of the  $n^{th}$  band for class  $i$ . This suggests that when the data has low correlation (close to zero), following transformation class separability is determined largely by individual feature separabilities and can be estimated by summing those single feature separabilities. Therefore, single band separability can be used as an approximate measure for feature selection from features that are poorly correlated.

Generally, high data variance is usually needed for separating different classes in an image and, thus, higher order principal components with small variances provide little significant information. Therefore, it is possible simply to select the first few high variance features and ignore the higher ordered principal components. However, it is important to recognise that some features selected in this way may be misleading. For example, original noisy bands will lead to some principal components with high variance but low separability.

## 13.7

### Regularised Covariance Estimators

Another approach that can be used to generate acceptable approximations to class covariance matrices is to make use of a process called regularisation, in which the poorly estimated class conditional covariance matrices are mixed with matrices that are known to be better determined, even if they are not class specific.

Let  $\Sigma_i$  be the estimate of the class covariance matrix obtained from the available training data for the class  $\omega_i$ . If there are not sufficient training samples available  $\Sigma_i$  will be a poor estimate. Let  $\Sigma_M$  be the covariance matrix computed from the full set of training samples – in other words it will be a global covariance matrix which reflects the scatter of the complete set of training data. Because this is based on a greater number of samples it is likely to be more accurate, for what it is, than the set of  $\Sigma_i$ .

Then an approximation that can be used for the class conditional covariance matrix is

$$\Sigma_i^{approx} = \alpha \Sigma_i + (1 - \alpha) \Sigma_M \quad (13.2)$$

where  $\alpha$  is a mixing parameter. Often diagonal versions of one of the constituent matrices would be used in (13.2), particularly for the original class covariance estimate.

Thus more often (13.2) would be

$$\Sigma_i^{approx} = \alpha \operatorname{diag} \Sigma_i + (1 - \alpha) \Sigma_M \quad (13.3a)$$

or

$$\Sigma_i^{approx} = \alpha \operatorname{trace}(\Sigma_i) I + (1 - \alpha) \Sigma_M \quad (13.3b)$$

The parameter  $\alpha$  needs to be determined to ensure that the approximation is as good as possible. One way to do that is to vary  $\alpha$  and then see how well the covariance estimate performs, either with the training data set or with a set of testing data. Often the Leave One Out method of Sect. 11.5.2 is used for this purpose.

Another covariance estimator commonly used is (Landgrebe, 2003)

$$\begin{aligned} \Sigma_i^{approx} &= (1 - \alpha) \operatorname{diag} \Sigma_i + \alpha \Sigma_i & 0 \leq \alpha \leq 1 \\ &= (2 - \alpha) \Sigma_i + (\alpha - 1) \Sigma_M & 1 < \alpha \leq 2 \\ &= (3 - \alpha) \Sigma_M + (\alpha - 2) \operatorname{diag} \Sigma_M & 2 < \alpha \leq 3 \end{aligned} \quad (13.4)$$

Again the optimum value for  $\alpha$  would be found by using the Leave One Out method on the training data.

It is interesting to examine the actual nature of this last estimate for some specific values of  $\alpha$ , noting the nature of the class conditional distributions that result, and the likely forms of the discriminant functions. For example:

- For  $\alpha = 0$ ,  $\Sigma_i^{approx} = \operatorname{diag} \Sigma_i$ , meaning that each class is represented by the diagonal elements of its class covariance matrix, and that cross correlations are ignored. Consequently, the classes are assumed to be distributed hyperelliptically with axes parallel to the spectral axes. A linear decision surface will result.
- For  $\alpha = 1$ ,  $\Sigma_i^{approx} = \Sigma_i$ , meaning that each class is represented by its actual class conditional covariance matrix, giving quadratic decision surfaces between the classes. This will give full multi-normal maximum likelihood classification.
- For  $\alpha = 2$ ,  $\Sigma_i^{approx} = \Sigma_M$ , meaning that all classes are assumed to have the same covariance matrix (equivalent to the global covariance), again generating linear decision surfaces.
- For  $\alpha = 3$ ,  $\Sigma_i^{approx} = \operatorname{diag} \Sigma_M$ , meaning again that all classes have the same covariance matrix, but in this case it consists just of the diagonal terms of the global covariance matrix. All class covariances will be identically hyperelliptical with axes parallel to the spectral axes, resulting in linear decision surfaces.

## 13.8 Compression of Hyperspectral Data

Owing to the large data volumes involved, storage and transmission of data from imaging spectrometers benefit from the application of procedures that will reduce data volume without substantially affecting the information content. Those procedures are generally in the form of codes that represent the spectra in reduced form. The binary codes of Sect. 13.4.3 are typical of codes that could be used, although

with such reductions in the spectra significant information loss (allowing the spectra to be used over a large number of applications) could be expected.

More sophisticated codes minimise information loss while compressing the data. The principal components transformation is an example. The higher order components with low variance can be discarded without significant information loss and yet with a reduction in storage requirement in proportion to the number of bands discarded. Also, the original spectral or image data can be reconstructed from the reduced representation (using an inverse principal components transform) although with loss of information. Sometimes the information loss is referred to as distortion since the reconstructed data will differ, depending on the level of loss of detail, from the original.

An alternative transformation widely used in the television and video industry is the Discrete Cosine Transform (Rao and Yip, 1990). The DCT is similar in principle to the Discrete Fourier Transform of Sect. 7.7, but with cosine expansion functions instead of complex exponentials as seen in (7.16).

If the user can tolerate substantial amounts of distortion then significant compression of remote sensing imagery is possible; figures as high as 100 times reduction in volume have been reported, but one is then led to question the integrity of the compressed data. Generally, those compression schemes that allow the original image to be reconstructed without error (so-called lossless compression algorithms) will give compression ratios of about 2 to 3.

A compression scheme well matched to the needs of remote sensing is referred to as vector quantisation, based upon the use of a so-called code book. That book contains a number of representative pixel vectors (for example class means) that could be obtained from training data, or possibly could even be prototypical reference spectra. Each code book vector is given a label (such as a number or even a class symbol).

Now imagine an image has to be transmitted over a telecommunications channel. If the spectrum matches exactly one of the stored spectra then only the label need be transmitted. The receiver also has a copy of the code book and can retrieve the spectrum in question through matching the label. If the spectrum does not match a code book entry exactly then transmitting the label of the nearest match will incur an error. Whether that error is acceptable, or whether a correction needs to be transmitted with the label of closest match, will depend on the application. The efficacy of the scheme depends upon how well the code book represents the range of pixel vectors in the image. A good code book will give rise to small differences (errors) between code book entries and pixel vectors to be transmitted. Such small differences can be encoded using a small number of bits (substantially smaller than the number of bits in the original pixel vector), so that good data compression is achieved.

A simple illustration is given in Table 13.2 in which 10 SPOT multispectral vectors are to be sent over a channel. Ordinarily, with each band represented by 8 bits, the ten pixels require  $10 \times 3 \times 8 = 240$  bits to be transmitted. However, recognising there are two clusters in the data and using the cluster means as code book vectors, it is possible to represent each of the pixels to be transmitted by their difference

**Table 13.2.** Simple illustration of vector quantisation

Cluster 1 (e.g. vegetation)					Cluster 2 (e.g. soil)				
Original pixel vectors:									
1	2	3	4	5	6	7	8	9	10
50	55	60	58	48	48	49	55	53	51
10	11	12	9	9	70	69	73	71	68
150	152	148	154	160	171	163	165	167	160

Code book entries (cluster means):									
—					—				
54					51				
10					70				
153					165				

Differences between pixel vectors and nearest code book entry:									
1	2	3	4	5	6	7	8	9	10
-4	1	6	4	-6	-3	-2	4	-2	0
0	1	2	-1	-1	0	-1	3	-1	-2
-3	-1	-5	1	7	6	-2	0	2	-5

(error) from the nearest mean. There are 8 distinct differences (between 0 and 7); they can be distinguished from each other (including sign) by allowing a 4 bit word for coding them. Thus the number of bits then to be transmitted is  $10 \times 3 \times 4 = 120$  bits, plus one bit per pixel to indicate the code book vector label (one bit is enough to represent just two labels – i.e. 0 or 1) and  $2 \times 3 \times 8 = 48$  bits to transmit the code book beforehand. Thus the vector quantised scheme requires  $120 + 10 + 48 = 178$  bits for the 10 pixels. The “compression ratio” is  $240/178 = 1.35$  with the ability to reconstruct the original pixel vectors without loss (distortion).

Further compression of the data is possible by using a more efficient coding process on the errors. Rather than simply allocating (in this example) 3 bits per difference (based on the observation that there are 8 different errors to transmit) shorter code words (in terms of numbers of bits) can be ascribed to the most commonly encountered errors (in this example 1 and 2). Details of this refinement, vector quantisation in general and the overall issue of compression in remotely sensed data can be found in Ryan and Arnold (1997a,b).

## 13.9

### Spectral Unmixing: End Member Analysis

A challenge that has faced interpreters throughout the history of remote sensing has been the need to handle mixed pixels – i.e. those pixels that represent a mixture of cover types or information classes. Several early studies attempted to resolve the proportions of pure cover types within mixed pixels by assuming that the measured radiance is a linear combination of the radiances of the “pure” constituents in each of the imaging wavebands used.

With low resolution (multispectral) data the approach generally did not meet with a great deal of success because most cover types are not well differentiated in the small number of wavebands used. However, with hyperspectral data, the prospect of uniquely characterising a vast number of earth cover types, and thus differentiating them from each other spectroscopically, suggests that the mixing approach should be re-visited as a means for establishing mixture proportions of pure cover types in pixels. This has particular relevance in minerals mapping where abundance maps for minerals of interest can then be produced based upon the proportions determined for all pixels in a given image.

The process can be developed mathematically in the following manner. Assume there are  $M$  pure cover types in the image of interest. In the nomenclature of mixing models these are referred to as *endmembers*. Let the proportions of the various endmembers in a pixel be represented by  $f_m, m = 1, \dots, M$ . These are the unknowns in the process which we wish to find, based on observation of the hyperspectral reflectance of the pixel.

Let  $R_n, n = 1, \dots, N$  be the observed reflectance of the pixel in the  $n^{th}$  spectral band of the sensor and  $a_{n,m}$  be the spectral reflectance in the  $n^{th}$  band of the  $m^{th}$  endmember. Then we assume

$$R_n = \sum_{m=1}^M f_m a_{n,m} + \xi_n \quad n = 1, \dots, N$$

where  $\xi_n$  is an error in band  $n$ . The equation says that the observed reflectance in each band is the linear sum of the reflectances of the endmembers; the extent to which that does not work exactly in a given situation is encapsulated in the error term.

An assumption that allows us to use linear mixing in this form is that the incident energy is scattered only once to the sensor from the landscape and does not undergo multiple scatterings among, for example, foliage components.

The above mixing equation can be expressed in matrix form as

$$\mathbf{R} = \mathbf{Af} + \boldsymbol{\xi}$$

where  $\mathbf{f}$  is a column vector of size  $M$ ,  $\mathbf{R}$  and  $\boldsymbol{\xi}$  are column vectors of size  $N$  and  $\mathbf{A}$  is an  $N \times M$  matrix of endmember spectral signatures (by column).

Spectral unmixing, as the process is called, involves finding a set of endmember proportions that will minimise the error vector  $\boldsymbol{\xi}$ . On the assumption that the correct set of endmembers has been chosen the problem then becomes one of solving the simpler equation

$$\mathbf{R} = \mathbf{A}\mathbf{f}$$

Normally there are more equations than unknowns so that simple inversion of the last equation to find the vector of mixing proportions is not possible. Instead, a least squares solution is found by using the pseudo inverse

$$\mathbf{f} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{R}.$$

It should be mentioned that there are two constraints that the mixing proportions of the endmembers are expected to satisfy. The first is that the proportions should sum to unity and the second is that they should all be non-negative:

$$\sum_{m=1}^M f_m = 1$$

$$0 \leq f_m \leq 1 \text{ for all } m$$

As discussed by Gross and Schott (1998) these constraints are sometimes violated if the endmembers are derived from average cover type spectra or the endmember selection is poor.

## References for Chapter 13

Information on radiometric correction of imaging spectrometer data using atmospheric and solar curve models can be found in Gao et al. (1993). Roberts et al. (1985, 1986) can be consulted for details of the empirical line and flat field methods of correction, while Green and Craig (1985) develop the Log Residuals technique in detail.

Piech and Piech (1987) have proposed a general method to detect all the absorption features in a pixel spectrum automatically and order them by their depth. The original spectrum is convolved with Gaussian masks over a range of standard deviations (called scale parameters). In so doing, absorption features disappear with increasing scale parameter and, as a result, a set of progressively smoothed spectra is obtained. Then, the derivative of the smoothed spectrum is calculated and the zero crossings indicate the local minima.

Clark (2003) develops this approach further, and uses expert systems to assist in the identification.

Binary coding of hyperspectral imagery is treated in Mazer et al. (1988), Vane and Goetz (1993), Jia (1996) and Jia and Richards (1993), while Jia (1996) and Jia and Richards (1994) cover the block based methods for treating data of such high dimensionality.

The problem of small training sets is treated in detail by Landgrebe (2003) who also demonstrates that unlabelled pixels can be used to supplement labelled training data in an endeavour to develop more reliable training statistics for Bayesian classification.

G.P. Anderson, J. Wang and J.H. Chetwynd, 1995: MODTRAN3: An Update and Recent Validations Against Airborne High Resolution Interferometer Measurements, 5th Annual JPL Airborne Sciences Workshop, Vol 1 (AVIRIS Workshop), 5–8.

R.N. Clark, G.A. Swayze, K.E. Livo, R.F. Kokaly, S.J. Sutley, J.B. Dalton, R.R. McDougal and C.A. Gent, 2003: Imaging Spectroscopy: Earth and Planetary Remote Sensing with the USGS Tetracorder and Expert Systems. *J. Geophysical Research*, 108, E12, 5.1–5.44.

- B.C. Gao, K.B. Heidebrecht and A.F.H. Goetz, 1992: Atmospheric Removal Program (ATREM) Users Guide, Centre for the Study of Earth from Space, University of Colorado, Boulder.
- B.C. Gao, K.B. Heidebrecht and, A.F.H. Goetz, 1993: Derivation of Scaled Surface Reflectance from AVIRIS Data, *Remote Sensing of Environment*, 44, 165–178.
- A.A. Green and M.D. Craig, 1985: Analysis of Aircraft Spectrometer Data with Logarithmic Residuals, Proc. AIS workshop, JPL Publication 85-41, Jet Propulsion Laboratory, Pasadena, California, April 8–10, 111–119.
- H.N. Gross and J.R. Schott, 1998. Application of Spectral Mixture Analysis and Image Fusion Techniques for Image Sharpening. *Remote Sensing of Environment*, 63, 85–94.
- X. Jia, 1996: Classification Techniques for Hyperspectral Remote Sensing Image Data, PhD Thesis, School of Electrical Engineering, University College, ADFA, The University of New South Wales.
- X. Jia and J.A. Richards, 1993: Binary Coding of Imaging Spectrometer Data for Fast Spectral Matching and Classification, *Remote Sensing of Environment*, 43, 47–53.
- X. Jia and J.A. Richards, 1994: Efficient Maximum Likelihood Classification for Imaging Spectrometer Data Sets, *IEEE Trans on Geoscience and Remote Sensing*, 32, 274–281.
- D.A. Landgrebe, 2003: *Signal Theory Methods in Multispectral Remote Sensing*, N.J., Wiley.
- F.A. Kruse, A.B. Letkoff, J.W. Boardman, K.B. Heidebrecht, A.T. Shapiro, P.J. Barloon and A.F.H. Goetz, 1993: The Spectral Image Processing System (SIPS) – interactive visualisation and analysis of imaging spectrometer data, *Remote Sensing of Environment*, 44, 145–163.
- A.S. Mazer, M. Martin, M. Lee and J.E. Dolomon, 1988: Image Processing Software for Imaging Spectrometry Data Analysis, *Remote Sensing of Environment*, 24, 201–211.
- M.A. Piech and K.R. Piech, 1987: Symbolic Representation of Hyperspectral Data, *Applied Optics*, 26, 4018–4026.
- K.R. Rao and P. Yip, 1990: *Discrete Cosine Transform*. New York: Academic.
- D.A. Roberts, Y. Yamaguchi and R.J.P. Lyon, 1985: Calibration of Airborne Imaging Spectrometer Data to Percent Reflectance Using Field Spectral Measurements, 19th International Symposium on Remote Sensing of Environment, Ann Arbor, Michigan, October 21–25.
- D.A. Roberts, Y. Yamaguchi and R.J.P. Lyon, 1986: Comparison of Various Techniques for Calibration of AIS Data, in Proceedings, 2nd AIS workshop, JPL Publication 86-35, Jet Propulsion Laboratory, Pasadena, California, 21–30.
- M.J. Ryan and J.F. Arnold, 1997a: The Lossless Compression of AVIRIS Images by Vector Quantization. *IEEE Trans Geoscience and Remote Sensing*, 35, 546–550.
- M.J. Ryan and J.F. Arnold, 1997b: Lossy Compression of Hyperspectral Data Using Vector Quantization. *Remote Sensing of Environment*, 61, 419–436.
- P.N. Slater, 1980: *Remote Sensing: Optics and Optical Systems*. Reading Mass: Addison-Wesley.
- P.H. Swain and S.M. Davis (eds.), 1978: *Remote Sensing: The Quantitative Approach*, New York: McGraw-Hill.
- G. Vane and A.F.H. Goetz, 1993: Terrestrial Imaging Spectrometry: Current Status, Future Trends, *Remote Sensing of Environment*, 44, 117–126.

## Problems

**13.1** The block based maximum likelihood classification scheme of Sect. 13.5.2 requires decisions to be taken about what blocks to use. From your knowledge of the spectral responses of

the three common ground cover types of vegetation, soil and water, recommend an acceptable set of block boundaries that might always be used with AVIRIS data. You may wish also to take note of the major water absorption features in AVIRIS spectra as seen in Fig. 1.9.

**13.2** Using the results of question 1 or otherwise discuss how the canonical analysis transformation might take advantage of partitioning the covariance matrices into blocks.

**13.3** Does partitioning the covariance matrix into blocks assist minimum distance classification?

**13.4** (a) Consider the block based approach to principal components analysis as developed in Sect. 13.6.2. Suppose several stages of transformation without feature reduction are used as depicted in Fig. 13.15. Prove that the overall data variance after the final transformation is the same as that generated had a single stage principal components transform been carried out.

(b) (This requires significant matrix analysis skills) As in part (a), but, a principal components transform without segmentation is finally performed on the data which are obtained after several stages of segmented principal components transform (without feature reduction) are used. Prove that the data variance of each feature is the same as that generated had conventional PCT been carried out for the original data.

**13.5** Consider the simple binary coding scheme (with one threshold) developed in Sect. 13.4.3.1 and illustrated in Fig. 13.10. How many distinct codes are possible for AVIRIS, TM and SPOT HRV data sets? Why would binary codes not be a sufficient representative form for SPOT and MSS data?

**13.6** Suppose a particular image contains just two cover types – vegetation and soil. A pixel identification exercise is carried out to attempt to attach either a soil or vegetation label to each pixel and thereby come up with an estimate of the proportion of vegetation in the region being imaged. From homogeneous regions of the image it is possible easily to label pure soil and pure vegetation pixels. Clearly the image also contains a number of mixed pixels and so end member analysis is considered as a means for resolving their soil/vegetation proportions. Is the additional work justified if the approximate proportion of vegetation to soil is 1:100, 50:50 or 100:1?

**13.7** Explain the concept of transmittance and name the main gases which cause markedly low atmospheric transmittance at wavelength(s) between 400–2400 nm.

**13.8** Describe briefly spectral library searching techniques, stating why they are feasible to use with imaging spectrometer data and noting their advantages over statistical classification methods.

**13.9** The principal components transform and the Bhattacharyya distance can both be used for band reduction. Comment on the main differences between the two methods for this application.

**13.10** Two-threshold coding is normally not recommended. Explain why.

**13.11** When three-threshold coding is used, make suggestions on how to define the three thresholds, particularly the upper and lower thresholds if the mean brightness value over the spectrum is used as the middle threshold.

**13.12** In the spectral angle mapper technique, the angle of a pixel vector in the spectral space needs to be determined (Fig. 13.8). Write down the formula for calculating the angle for the general case with  $N$  bands.

# **Appendix A**

## **Missions and Sensors**

This appendix contains descriptive and technical information on satellite and aircraft missions and the characteristics of their sensors. It commences by looking briefly at those programs intended principally for gathering weather information, and proceeds to missions for earth observational remote sensing, including hyperspectral and radar platforms and sensors.

Sufficient detail is given on data characteristics so that implications for image processing and analysis can be understood. In most cases mechanical and signal handling properties are not given, except for a few historical and illustrative cases.

### **A.1**

#### **Weather Satellite Sensors**

##### **A.1.1**

##### **Polar Orbiting and Geosynchronous Satellites**

Two broad types of weather satellite are in common use. One is of the polar orbiting, or more generally low earth orbit, variety whereas the other is at geosynchronous altitudes. The former typically have orbits at altitudes of about 700 to 1500 km whereas the geostationary altitude is approximately 36,000 km (see Appendix B). Typical of the low orbit satellites are the current NOAA series (also referred to as Advanced TIROS-N, ATN), and their forerunners the TIROS, TOS and ITOS satellites. The principal sensor of interest from this book's viewpoint is the NOAA AVHRR. This is described in Sect. A.1.2 following.

The Nimbus satellites, while strictly test bed vehicles for a range of meteorological and remote sensing sensors, also orbited at altitudes of around 1000 km. Nimbus sensors of interest include the Coastal Zone Colour Scanner (CZCS) and the Scanning Multichannel Microwave Radiometer (SMMR). Only the former is treated below.

Geostationary meteorological satellites have been launched by the United States, Russia, India, China, ESA and Japan. These are placed in equatorial geosynchronous orbits.

### A.1.2

#### **The NOAA AVHRR (Advanced Very High Resolution Radiometer)**

The AVHRR has been designed to provide information for hydrologic, oceanographic and meteorologic studies, although data provided by the sensor does find application also to solid earth monitoring. An earlier version of the AVHRR contained four wavelength bands. Table A.1 however lists the bands available on the current generation of instrument (NOAA 17).

**Table A.1.** NOAA advanced very high resolution radiometer

Spatial resolution	1.1 km at nadir
Dynamic range	10 bit
Swath width	2399 km
Spectral bands:	
channel 1	0.58 – 0.68 µm
channel 2	0.725 – 1.0 µm
channel 3	3.55 – 3.93 µm
channel 3a	1.58 – 1.64 µm
channel 4	10.3 – 11.3 µm
channel 5	11.5 – 12.5 µm

### A.1.3

#### **The Nimbus CZCS (Coastal Zone Colour Scanner)**

The CZCS was a mirror scanning system, carried on Nimbus 7, designed to measure chlorophyll concentration, sediment distribution and general ocean dynamics including sea surface temperature. Its characteristics are summarised in Table A.2.

**Table A.2.** Nimbus coastal zone colour scanner

Spatial resolution	825 m at nadir
Dynamic range	8 bit
Swath width	1566 km
Spectral bands:	
channel 1	0.433 – 0.453 µm
channel 2	0.510 – 0.530 µm
channel 3	0.540 – 0.560 µm
channel 4	0.660 – 0.680 µm
channel 5	0.700 – 0.800 µm
channel 6	10.5 – 12.5 µm

**A.1.4****GMS VISSR (Visible and Infrared Spin Scan Radiometer) and GOES Imager**

Geostationary meteorological satellites such as GMS (Japan) and the earlier GOES (USA) are spin stabilized with their spin axis oriented almost north-south. The primary sensor on these, the VISSR, scans the earth's surface by making use of the satellite spin to acquire one line of image data (as compared with an oscillating mirror in the case of AVHRR, CZCS, MSS and TM sensors), and by utilizing a stepping motor to adjust the angle of view on each spin to acquire successive line of data (on orbiting satellites it is the motion of the vehicle relative to the earth that displaces the sensor between successive scan lines). The characteristics of the VISSR are summarised in Table A.3.

The most recent GOES environmental satellites are 3 axis stabilised and carry a GOES Imager with characteristics as shown in Table A.3.

**Table A.3.** VISSR and GOES Imager characteristics

	Band ( $\mu\text{m}$ ) at nadir (km)	Spatial resolution (bits)	Dynamic range
VISSR	0.55 – 0.90 (visible)	1.25	6
	6.7 – 7.0	5	8
	10.5 – 11.5	5	8
	11.5 – 12.5 (thermal infrared)	5	8
GOES Imager	0.55 – 0.75	1	10
	3.80 – 4.00	1	10
	6.50 – 7.00	1	10
	10.20 – 11.20	1	10
	11.50 – 12.50	1	10

**A.2****Earth Resource Satellite Sensors  
in the Visible and Infrared Regions****A.2.1****The Landsat System**

The Landsat earth resources satellite system was the first designed to provide near global coverage of the earth's surface on a regular and predictable basis.

The first three Landsats had identical orbit characteristics, as summarised in Table A.4. The orbits were near polar and sun synchronous – i.e., the orbital plane precessed about the earth at the same rate that the sun appears to move across the

**Table A.4.** Landsat 1, 2, 3 orbit characteristics

Orbit:	Sun synchronous, near polar; nominal 9:30 am descending equatorial crossing; inclined at about 99° to the equator
Altitude:	920 km (570 mi)
Period:	103 min
Repeat Cycle:	14 orbits per day over 18 days (251 revolutions)

face of the earth. In this manner data was acquired at about the same local time on every pass.

All satellites acquired image data nominally at 9:30 a.m. local time on a descending (north to south) path; in addition Landsat 3 obtained thermal data on a night-time ascending orbit for the few months that its thermal sensor was operational. Fourteen complete orbits were covered each day, and the fifteenth, at the start of the next day, was 159 km advanced from orbit 1, thus giving a second day coverage contiguous with that of the first day. This advance in daily coverage continued for 18 days and then repeated. Consequently complete coverage of the earth's surface was given, with 251 revolutions in 18 days.

The orbital characteristics of the second generation Landsats, commencing with Landsats 4 and 5, are different from those of their predecessors. Again image data is acquired nominally at 9:30 a.m. local time in a near polar, sun synchronous orbit; however the spacecraft are at the lower altitude of 705 km. This lower orbit gives a repeat cycle of 16 days at 14.56 orbits per day. This corresponds to a total of 233 revolutions every cycle. Table A.5 summarises the Landsat 4, 5 orbit characteristics. Unlike the orbital pattern for the first generation Landsats, the day 2 ground pattern for Landsats 4 and 5 is not adjacent and immediately to the west of the day 1 orbital pattern. Rather it is displaced the equivalent of 7 swath centres to the west. Over 16 days this leads to the repeat cycle.

Landsat 6, launched in 1993, was not successfully placed in orbit and was lost over the Atlantic Ocean. Landsat 7 is a similar satellite in all respects.

Whereas Landsats 1, 2 and 3 contained on-board tape recorders for temporary storage of image data when the satellites were out of view of earth stations, Landsats 4 and 5 do not, and depend on transmission either to earth stations directly or via the geosynchronous communication satellite TDRS (Tracking and Data Relay Satellite). TDRS is a high capacity communication satellite that is used to relay data from a

**Table A.5.** Orbit parameters for Landsats 4, 5, 7

Orbit:	Near polar, sun synchronous; nominal 9:30 am descending equatorial crossing (10:00 am for Landsat 7)
Altitude:	705 km
Period:	98.9 min
Repeat Cycle:	14.56 orbits per day over 16 days (total of 233 revolutions)

number of missions, including the Space Shuttle. Its ground receiving station is in White Sands, New Mexico from which data is relayed via domestic communication satellites. Landsat 7 also uses TDRS for data downlinking but has an on-board solid state recorder for temporary storage.

### A.2.2 The Landsat Instrument Complement

Three imaging instruments have been used with the Landsat satellites to date. These are the Return Beam Vidicon (RBV), the Multispectral Scanner (MSS) and the Thematic Mapper (TM). Table A.6 shows the actual imaging payload for each satellite along with historical data on launch and out-of-service dates. Two different RBV's were used: a multispectral RBV package was incorporated on the first two satellites, while a panchromatic instrument with a higher spatial resolution was used on Landsat 3. The MSS on Landsat 3 also contained a thermal band; however this operated only for a few months.

**Table A.6.** Landsat payloads, launch and out of service dates

Satellite	Imaging Instruments		Launched	Out-of-service
Landsat 1	RBV <sup>m</sup>	MSS	23 Jul 1972	6 Jan 1978
Landsat 2	RBV <sup>m</sup>	MSS	22 Jan 1975	27 Jul 1983
Landsat 3	RBV <sup>p</sup>	MSS <sup>t</sup>	5 Mar 1978	7 Sept 1983
Landsat 4	MSS	TM	16 Jul 1982	Aug 1993
Landsat 5	MSS	TM	1 Mar 1984	-
Landsat 6		ETM	lost on launch	-
Landsat 7		ETM+	15 Apr 1999	-

m – multispectral RBV  
 p – panchromatic RBV  
 t – MSS with thermal band

The MSS was not used after Landsat 5. With the launch of Landsat 7 an Enhanced Thematic Mapper + (ETM+) was added.

The following sections provide an overview of the three Landsat instruments, especially from a data characteristic point-of-view.

### A.2.3 The Return Beam Vidicon (RBV)

As the name suggests the RBV's were essentially television camera-like instruments that took "snapshot" images of the earth's surface along the ground track of the satellite. Image frames of 185 km × 185 km were acquired with each shot, repeated at 25 s intervals to give contiguous frames in the along track direction at the equivalent ground speed of the satellite.

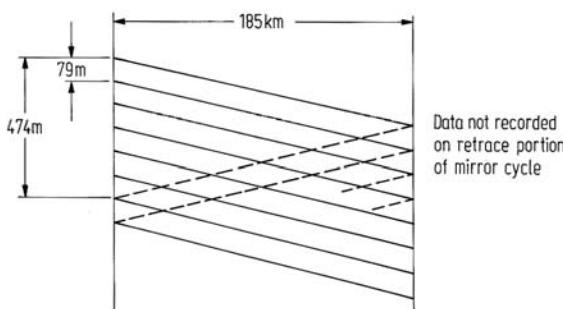
Three RBV cameras were used on Landsats 1 and 2, distinguished by different transmission filters that allowed three spectral bands of data to be recorded as shown in Table A.7. On Landsat 3 two RBV cameras were used; however both operated panchromatically and were focussed to record data swaths of 98 km, overlapped to give a total swath of about 185 km. By so doing a higher spatial resolution of 40 m was possible, by comparison to 80 m for the earlier RBV system.

Historically the spectral ranges recorded by the RBV's on Landsats 1 and 2 were referred to as bands 1, 2 and 3. The MSS bands (see following) in the first generation of Landsats were numbered to follow on in this sequence.

#### A.2.4

#### The Multispectral Scanner (MSS)

The Multispectral Scanner was the principal sensor on Landsats 1, 2 and 3 and was the same on each spacecraft with the exception of an additional band on Landsat 3. The MSS is a mechanical scanning device that acquires data by scanning the earth's surface in strips normal to the satellite motion. Six lines are swept simultaneously by an oscillating mirror and the reflected solar radiation so monitored is detected in four wavelength bands for Landsats 1 and 2, and five bands for Landsat 3, as shown in Table A.7. A schematic illustration of the six line scanning pattern used by the MSS is shown in Fig. A.1. It is seen that the sweep pattern gives rise to an MSS swath width of 185 km thereby corresponding to the image width of the RBV. The width of each scan line corresponds to 79 m on the earth's surface so that the six lines simultaneously correspond to 474 m. Approximately 390 complete six-line scans are collected to provide an effective image that is also 185 km in the along track direction. For Landsats 1 and 2, 24 signal detectors were required to provide four spectral bands from each of the six scan lines. A further two were added for the thermal band data of Landsat 3. Those detectors are illuminated by radiation reflected from the oscillating scanning mirror in the MSS, and produce a continuously varying



**Fig. A.1.** The six line scanning pattern used by the Landsat multispectral scanner. Dimensions are in equivalent measurements on the ground. This scanning pattern is the same in each of bands 4 to 7. The same six line pattern is used on Landsats 4 and 5 except that the strip width is 81.5 m and 82.5 m respectively

**Table A.7.** Characteristics of the Landsat imaging devices

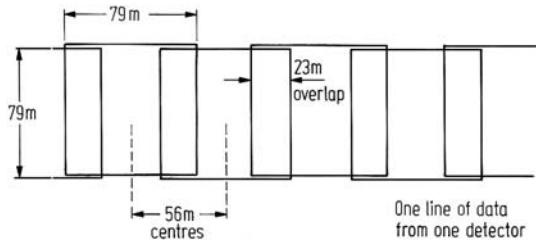
Instrument	Spectral bands (μm)		IFOV (m)	Dynamic range (bits)
RBV <sup>m</sup>	1.	0.475– 0.575 (blue)	79 × 79	
	2.	0.580– 0.680 (red)	79 × 79	
	3.	0.689– 0.830 (near IR)	79 × 79	
RBV <sup>p</sup>		0.505– 0.750 (panchromatic)		40 × 40
MSS	4. <sup>a</sup>	0.5 – 0.6 (green)	79 × 79	7
	5.	0.6 – 0.7 (red)	79 × 79	7
	6.	0.7 – 0.8 (near IR)	79 × 79	7
	7.	0.8 – 1.1 (near IR)	79 × 79	6
	8. <sup>b</sup>	10.4 – 12.6 (thermal)	237 × 237	
TM	1.	0.45 – 0.52 (blue)	30 × 30	8
	2.	0.52 – 0.60 (green)	30 × 30	8
	3.	0.63 – 0.69 (red)	30 × 30	8
	4.	0.76 – 0.90 (near IR)	30 × 30	8
	5.	1.55 – 1.75 (mid IR)	30 × 30	8
	7. <sup>c</sup>	2.08 – 2.35 (mid IR)	30 × 30	8
	6.	10.4 – 12.5 (thermal)	120 × 120	8
ETM+	1.	0.450– 0.515 (blue)	30 × 30	8
	2.	0.525– 0.605 (green)	30 × 30	8
	3.	0.630– 0.690 (red)	30 × 30	8
	4.	0.775– 0.900 (near IR)	30 × 30	8
	5.	1.550– 1.750 (mid IR)	30 × 30	8
	7.	2.090– 2.350 (mid IR)	30 × 30	8
	6.	10.40 – 12.50 (thermal)	60 × 60	8
	pan	0.520– 0.900	13 × 15	8

<sup>a</sup> MSS bands 4 to 7 have been renumbered MSS bands 1 to 4 from Landsat 4 onwards. IFOV = 81.5, 82.5 m for Landsats 4, 5.

<sup>b</sup> MSS band 8 was used only on Landsat 3.

<sup>c</sup> TM band 7 is out of sequence since it was added last in the design after the previous six bands had been firmly established. It was incorporated at the request of the geological community owing to the importance of the 2 μm region in assessing hydrothermal alteration.

electrical signal corresponding to the energy received along the 79 m wide associated scan line. The optical aperture of the MSS and its detectors for bands 4 to 7 is such that at any instant of time each detector sees a pixel that is 79 m in size also along the scan line. Consequently the effective pixel size (or instantaneous field of view IFOV) of the detectors is 79 m × 79 m. At a given instant the output from a detector is the integrated response from all cover types present in a 79 m × 79 m region of the earth's surface. Without any further processing the signal from the detector would appear to be varying continuously with time. However it is sampled in time to produce discrete measurements across a scan line. The sampling rate corresponds to pixel centres of 56 m giving a 23 m overlap of the 79 m × 79 m pixels, as depicted in Fig. A.2. The thermal infrared band on Landsat 3, band 8, has an IFOV of 239 m × 239 m. As a



**Fig. A.2.** The relationship between instantaneous field of view and pixel overlap for Landsat MSS pixels

result there are only two band 8 scan lines corresponding to the six for bands 4 to 7, as indicated above.

The IFOV's of the multispectral scanners on Landsats 4 and 5 have been modified to 81.5 m and 82.5 m respectively although the pixel centre spacing of 56 m has been retained. In addition the bands have been renamed as bands 1, 2, 3 and 4, corresponding to bands 4, 5, 6 and 7 from the earlier missions.

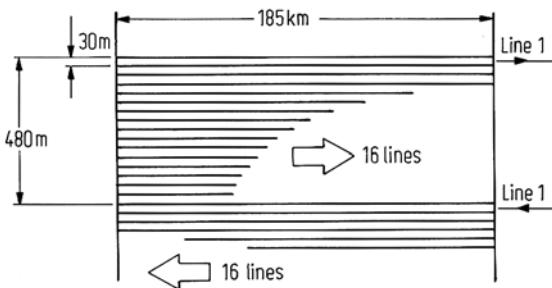
After being spatially sampled, the data from the detectors is digitised in amplitude into 6 bit words. Before so-encoding, the data for bands 4, 5 and 6 is compressed allowing decompression into effective 7 bit words upon reception at a ground station.

### A.2.5

#### The Thematic Mapper (TM) and Enhanced Thematic Mapper + (ETM+)

The Thematic Mapper is a mechanical scanning device as for the MSS, but has improved spectral, spatial and radiometric characteristics. Seven wavelength bands are used, with coverage as shown in Table A.7. Note that band 7 is out of place in the progression of wavelengths, it having been added, after the initial planning phase, at the request of the geological community. The Enhanced Thematic Mapper + carried on Landsat 7 includes a panchromatic band and improved spatial resolution on the thermal band.

Whereas the MSS of all Landsats scans and obtains data in one direction only, the TM acquires data in both scan directions again with a swath width of 185 km. Sixteen scan lines are swept simultaneously giving a 480 m strip across the satellite path, as illustrated in Fig. A.3. This permits a lower mirror scan rate compared with the MSS and thus gives a higher effective dwell time for a given spot on the ground, making possible the higher spatial resolution and improved dynamic range.



**Fig. A.3.** Scanning characteristics of the Landsat Thematic Mapper

### A.2.6

#### The SPOT HRV, HRVIR, HRG, HRS and Vegetation Instruments

The early French SPOT satellites (Système pour d’Observation de la Terre), carried two imaging devices referred to as HRV’s. These instruments utilize a different technology for image acquisition from that employed in the Landsat MSS and TM devices. Rather than using oscillating mirrors to provide cross-track scanning during the forward motion of the space platform, the SPOT HRV instruments consist of a linear array of charge coupled device (CCD) detectors. These form what is commonly referred to as a “push broom” scanner. Each detector in the array scans a strip in the along track direction. By having several thousand such detectors a wide swath can be imaged without the need for mechanical scanning. Moreover, owing to the long effective dwell time this allows for each pixel, a higher spatial resolution is possible. A trade-off however is that charge coupled device technology was not available for wavelengths into the middle infrared range at the time of early SPOT development. Consequently the spectral bands provided by the HRV are not unlike those of the Landsat MSS.

The HRV covers a ground swath width of 60 km; two instruments are mounted side by side in the spacecraft to give a total swath width of 117 km, there being a 3km overlap of the individual swaths.

Two imaging modes are possible. One is a multispectral mode and the other panchromatic. The imaging characteristics of these are summarised, along with the satellite orbital properties, in Table A.8.

An interesting property of the HRV is that it incorporates a steerable mirror to allow imaging to either side of nadir. This allows daily coverage for a short period along with a stereoscopic viewing capability.

SPOT 4, launched in March 1998, carries two instruments – the HRVIR (High Resolution Visible and Infrared) and the Vegetation instrument. Characteristics of both are summarised in Table A.8.

SPOT 5 carries an instrument known as HRG (High Resolution Geometry) that uses essentially the same wavebands as the HRVIR but with a higher spatial resolution. Its characteristics are also summarised in Table A.8. SPOT 5 also carries the Vegetation instrument, along with a new device called the HRS (High Resolution

**Table A.8.** Spot satellite and sensor characteristics

Orbit:	Near polar, sun synchronous; nominal 10.30 am descending equatorial crossing		
Altitude:	832 km		
Period:	101 min		
Repeat Cycle:	26 days		
Satellite	Imaging Instrument	Launched	Out-of-Service
SPOT 1	HRV ( $\times 2$ )	22 Feb 1986 reactivated 9 Jan 1997	5 Jan 1991
SPOT 2	HRV ( $\times 2$ )	22 Jan 1990	
SPOT 3	HRV ( $\times 2$ )	26 Sep 1993	14 Nov 1996
SPOT 4	HRVIR ( $\times 2$ ), Vegetation	24 Mar 1998	
SPOT 5	HRG ( $\times 2$ ), HRS, Vegetation	4 May 2002	
Instrument	Spectral bands ( $\mu\text{m}$ )	IFOV (m)	Swath (km)
HRV <sup>m</sup>	0.50 – 0.59 (green) 0.61 – 0.68 (red) 0.79 – 0.89 (near IR)	20 $\times$ 20 20 $\times$ 20 20 $\times$ 20	60 60 60
HRV <sup>p</sup>	0.51 – 0.73	10 $\times$ 10	60
HRVIR <sup>m</sup>	0.50 – 0.59 (green) 0.61 – 0.68 (red) 0.78 – 0.89 (near IR) 1.58 – 1.75 (mid IR)	20 $\times$ 20 20 $\times$ 20 20 $\times$ 20 20 $\times$ 20	60 60 60 60
HRVIR <sup>p</sup>	0.61 – 0.68	10 $\times$ 10	60
HRG <sup>m</sup>	0.50 – 0.59 (green) 0.61 – 0.68 (red) 0.79 – 0.89 (near IR) 1.58 – 1.75 (mid IR)	10 $\times$ 10 10 $\times$ 10 10 $\times$ 10 20 $\times$ 20	60 60 60 60
HRG <sup>p</sup>	0.49 – 0.69	5 $\times$ 5 and 2.5 $\times$ 2.5	60
Vegetation	0.45 – 0.52 (blue) 0.61 – 0.68 (red) 0.78 – 0.89 (near IR) 1.58 – 1.75 (mid IR)	1000 $\times$ 1000 1000 $\times$ 1000 1000 $\times$ 1000 1000 $\times$ 1000	2250 2250 2250 2250
HRS	0.49 – 0.69	5 $\times$ 10	120

<sup>m</sup> multispectral mode<sup>P</sup> panchromatic mode

Stereoscopy) that images fore and aft of the spacecraft to allow stereoscopic products to be developed.

### A.2.7

#### ADEOS (Advanced Earth Observing Satellite)

ADEOS-I was launched by the Japanese space agency NASDA in August 1996. It carried a number of imaging instruments and non-imaging sensors, including OCTS (Ocean Colour and Temperature Sensor), AVNIR (Advanced Visible and Near Infrared Radiometer), NSCAT (NASA Spectrometer), TOMS (Total Ozone Mapping Spectrometer), POLDER (Polarization and Directionality of the Earth's Reflectance),

**Table A.9.** ADEOS satellite and sensor characteristics

Orbit:	near polar, sun synchronous; nominal 10.30 am equatorial crossing			
Altitude:	797 km			
Period:	101 min			
Repeat Cycle:	41 days (ADEOS 2 has a 4 day repeat cycle)			
Instrument	Spectral bands ( $\mu\text{m}$ )	IFOV (m)	Swath (km)	Dynamic range (bits)
AVNIR <sup>m</sup>	0.42 – 0.50 0.52 – 0.60 0.61 – 0.69 0.76 – 0.89	16 × 16 16 × 16 16 × 16 16 × 16	80 80 80 80	8 8 8 8
AVNIR <sup>p</sup>	0.52 – 0.69	8 × 8	80	7
OCTS	0.412 ± 0.01 0.443 ± 0.01 0.490 ± 0.01 0.520 ± 0.01 0.565 ± 0.01 0.670 ± 0.01 0.765 ± 0.02 0.865 ± 0.02 3.55 – 3.88 8.25 – 8.80 10.3 – 11.4 11.4 – 12.7	700 × 700* 700 × 700 700 × 700	1400 1400 1400 1400 1400 1400 1400 1400 1400 1400 1400 1400	10 10 10 10 10 10 10 10 10 10 10 10
GLI	0.375 – 12.5 (36 bands @ 10 nm bandwidth)	1000 × 100 or 250 × 250	1600	12

IMG (Interferometric Monitor for Greenhouse Gases), ILAS (Improved Limb Atmospheric Sensor) and RIS (Retroreflector in Space). Its successor, ADEOS-II, was launched on 14 December 2002. The sensors on ADEOS-II (now also called MIDORI-II) are the GLI (Global Imager), AMSR (Advanced Microwave Scanning Radiometer), ILAS-II, POLDER and SeaWINDS.

Characteristics of the OCTS, GLI and AVNIR are given in Table A.9, along with spacecraft orbital details.

### A.2.8

#### Sea-Viewing Wide Field of View Sensor (SeaWiFS)

In August 1997 the OrbView-2 (SeaStar) satellite was launched, carrying the SeaWiFS sensor with characteristics as shown in Table A.10. Its wavebands have been chosen with ocean-related applications in mind.

**Table A.10.** SeaStar satellite and SeaWiFS sensor characteristics

Orbit:	near polar, sun synchronous			
Altitude:	705 km			
Period:	98.9 min			
Repeat Cycle:	1 day			
Instrument	Spectral bands ( $\mu\text{m}$ )	IFOV (m)	Swath (km)	Dynamic range (bits)
SeaWiFS	0.402 – 0.422 0.433 – 0.453 0.480 – 0.500 0.500 – 0.520 0.545 – 0.565 0.660 – 0.680 0.745 – 0.785 0.845 – 0.885	1100 × 1100 1100 × 1100	2800 2800 2800 2800 2800 2800 2800 2800	10 10 10 10 10 10 10 10

**Table A.11.** MOS orbit and sensor characteristics

MOS:	Altitude	908 km
	Orbit	sun synchronous, 99.1° inclination
		10–11 am equatorial crossing
	Repeat Cycle	17 days
MESSR:	Bands	0.51–0.59 $\mu\text{m}$ 0.61–0.69 $\mu\text{m}$ 0.73–0.80 $\mu\text{m}$ 0.80–1.10 $\mu\text{m}$
	IFOV	50 m × 50 m
	Dynamic range	8 bit
	Swath per	
	MESSR	100 km
VTIR:	Bands	0.5– 0.7 $\mu\text{m}$ 6.0– 7.0 $\mu\text{m}$ 10.5–11.5 $\mu\text{m}$ 11.5–12.5 $\mu\text{m}$
	IFOV	900 m × 900 m for visible channel 2700 m × 2700 m for the others
	Dynamic range	8 bit
	Swath Width	1500 km

### A.2.9

#### Marine Observation Satellite (MOS)

The Marine Observation Satellites MOS-1 and MOS-1b were launched by Japan in February 1987 and February 1990 respectively and were taken out of service in March 1995 and April 1996 respectively. While intended largely for oceanographic studies, the data from the satellites' two optical imaging sensors – the MESSR (Multispectrum Electronic Self Scanning Radiometer) and the VTIR (Visible and Thermal Infrared

Radiometer) are of value to land based remote sensing as well. The satellites also carried a Microwave Scanning Radiometer (MSR) intended for water vapour, snow and ice studies. Two MESSRs were used to provide side by side observations. Each has a 100 km swath width; with an overlap in coverage of 15 km the total available swath is 185 km.

Orbital details of the MOS satellites and characteristics of their optical sensors are given in Table A.11.

### A.2.10

#### **Indian Remote Sensing Satellite (IRS)**

A series of remote sensing satellites has been launched by India since March 1988. They carry imaging systems known as the LISS (Linear Imaging Self Scanner), the WiFS (Wide Field Sensor), the advanced version AWIFS, the Ocean Colour Monitor (OCM), the Multifrequency Scanning Microwave Radiometer (MSMR), the Molecular Optoelectronic Scanner (MOS) and a panchromatic sensor. Orbital details of the satellites and characteristics of the sensors are summarised in Table A.12.

### A.2.11

#### **RESURS-O1**

Russia has orbited a series of remote sensing satellites since 1985 under the name RESURS-O1. Table A.13 gives platform and sensor characteristics for the third in the series. The principal sensor, from which commercially available imagery is produced, is the MSU-SK, which is a conically scanning instrument.

### A.2.12

#### **The Earth Observing 1 (EO-1) Mission**

EO-1 was launched on 21 November 2000 into the same orbit as Landsat 7, but one minute behind, allowing near simultaneous, partly overlapping coverage. The Terra platform (see Sect. A.2.13) is essentially also in the same orbit, but 30 minutes behind EO-1. The two imaging instruments of importance on EO-1 are the Advanced Land Imager (ALI) and Hyperion, the characteristics of which are given in Table A.14.

### A.2.13

#### **Aqua and Terra**

The Aqua and Terra platforms are part of NASA's Earth Observing System. They were launched respectively on 4 May 2002 and 18 December 1999 in sun synchronous orbits comparable to those for Landsat 7, but with descending equatorial crossings of 1:30 am (or 1:30 pm ascending) for Aqua and around 10:30 am for Terra. They are also known as the Earth Observing System (EOS) PM (Aqua) and EOS AM platforms (Terra).

**Table A.12.** IRS satellite and sensor characteristics

Orbit:	Near polar, sun synchronous; nominal 10.35 am equatorial crossing			
Altitude:	904 km (1A, 1B), 817 km (1C), 736/825 km (1D)			
Period:	101 min			
Repeat Cycle:	22 days (1A, 1B), 24 days (1C, 1D)			
Satellite	imaging Instrument	Launched	Out-of-Service	
IRS-1A	LISSI	Mar 1988		
IRS-1B	LISSII	Aug 1991		
IRS-P2	LISSIII	16 Oct 1994		
IRS-1C	LISSIII, WiFS			
IRS-1D	LISSIII, Pan WiFS	29 Sep 1997		
IRS-P3	WiFS, MOS	21 Mar 1996		
IRS-P4	OCM, MSMR	May 1999	(Oceansat 1)	
IRS-P5	Pan	Scheduled 2004	(Cartosat 1)	
IRS-P6	LISSIII, LISS IV, AWiFS	17 Oct 2003	(Resourcesat 1)	
Instrument	Spectral bands ( $\mu\text{m}$ )	IFOV (m)	Swath (km)	
LISSI, II	0.45 – 0.52 0.52 – 0.59 0.62 – 0.68 0.77 – 0.86	73 $\times$ 73 (LISSI) 36 $\times$ 36 (LISSII)	146 146 146 146	7 7 7 7
LISSIII	0.52 – 0.59 0.62 – 0.68 0.77 – 0.86 1.55 – 0.59	23 $\times$ 23 23 $\times$ 23 23 $\times$ 23 70 $\times$ 70	142 – 146 142 – 146 142 – 146 142 – 146	7 7 7 7
Pan	0.5 – 0.57	10 $\times$ 10	70	7
WiFS	0.62 – 0.68 0.77 – 0.86	188 $\times$ 188 188 $\times$ 188	774 774	7 7
LISSIV	0.53 – 0.59 0.62 – 0.68 0.77 – 0.86	5.8 $\times$ 5.8 5.8 $\times$ 5.8 5.8 $\times$ 5.8	23.9 (XS mode) 70.3 (Pan mode)	7 7 7
AWiFS	0.52 – 0.59 0.62 – 0.68 0.77 – 0.86 1.55 – 1.70	56 $\times$ 56 56 $\times$ 56 56 $\times$ 56 56 $\times$ 56	740 740 740 740	10 10 10 10
OCM	0.4 – 0.885 (8 bands @ 20 nm bandwidth)	360 $\times$ 360	1420	12
MOS-A	0.756 – 0.768 (4 bands @ 1.4 nm bandwidth)	1570 $\times$ 1400	195	16
-B	0.408 – 1.015 (13 bands @ 10 mm bandwidth)	520 $\times$ 520	200	16
-C	1.600 (1 band @ 100 mm bandwidth)	520 $\times$ 640	192	16

**Table A.13.** RESURS-O1-3 satellite and sensor characteristics

Orbit:	Sun synchronous, circular		
Altitude:	678 km		
Period:	98 min		
Repeat Cycle:	21 days		
Instrument	Spectral bands (μm)	IFOV (m)	Swath (km)
MSU-SK	0.5 – 0.6 0.6 – 0.7 0.7 – 0.8 0.8 – 1.1 10.4 – 12.6	160 160 160 160 600	600 600 600 600 600

**Table A.14.** EO-1 sensor characteristics

Instrument	Spectral Bands (μm)	IFOV (m)	Swath (km)	Dynamic Range (bits)
Hyperion	0.4–2.4 (220 bands @ 10 nm bandwidth)	30 × 30	7.7	12
ALI	0.433–0.453	30 × 30	37	12
	0.450–0.515	30 × 30	37	12
	0.525–0.606	30 × 30	37	12
	0.639–0.690	30 × 30	37	12
	0.775–0.805	30 × 30	37	12
	0.845–0.890	30 × 30	37	12
	1.200–1.300	30 × 30	37	12
	1.550–1.750	30 × 30	37	12
	2.080–2.350	30 × 30	37	12
	0.480–0.690 (panchromatic)	10 × 10	37	12

The principal instruments on Terra are MODIS (Moderate Resolution Imaging Spectrometer), ASTER (Advanced Spaceborne Thermal Emission and Reflection Spectrometer), CERES (Clouds and the Earth's Radiant Energy System), MISR (Multi-angle Imaging SpectroRadiometer) and MOPITT (Measurement of Pollution in the Troposphere). The instrument complement on Aqua includes MODIS, a set of optical and microwave atmospheric sounders, CERES, and a scanning microwave radiometer. The characteristics of MODIS and ASTER are given in Table A.15.

**Table A.15.** Aqua and Terra sensor characteristics

Instrument	Spectral Bands (μm)	IFOV (m)	Swath (km)	Dynamic Range (bits)
MODIS*	0.620–0.670	250 × 250	2330	12
	0.841–0.876	250 × 250	2330	12
	0.459–2.155 (5 bands)	500 × 500	2330	12
	0.405–14.385 (29 bands)	1000 × 1000	2330	12
ASTER	0.52–0.60	15 × 15	60	8
	0.63–0.69	15 × 15	60	8
	0.76–0.86	15 × 15	60	8
	0.76–0.86 (backward looking)	15 × 15	60	8
	1.600–1.700	30 × 30	60	8
	2.145–2.185	30 × 30	60	8
	2.185–2.225	30 × 30	60	8
	2.235–2.285	30 × 30	60	8
	2.295–2.365	30 × 30	60	8
	2.360–2.430	30 × 30	60	8
	8.125–8.475	90 × 90	60	12
	8.475–8.825	90 × 90	60	12
	8.925–9.275	90 × 90	60	12
	10.250–10.950	90 × 90	60	12
	10.950–11.650	90 × 90	60	12

\* The band description for MODIS is quite complex, since groups of bands are targeted on specific applications. Full details can be obtained from the MODIS home page at <http://modis.gsfc.nasa.gov>

**A.2.14****Ikonos**

The Ikonos satellite was launched on 24 September 1999. It is in a sun-synchronous, near polar orbit at an altitude of 681 km, with a repeat cycle of 35 days (although a 1.5 day revisit capacity is possible with off-nadir pointing). It has a descending equatorial crossing of 10:30 am, comparable to SPOT. Characteristics of its imaging sensor are given in Table A.16.

**Table A.16.** Ikonos sensor characteristics

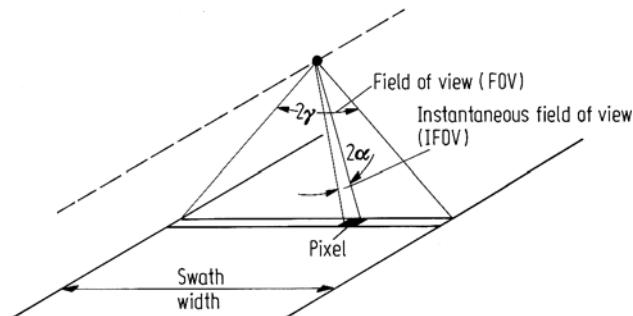
	Spectral Bands ( $\mu\text{m}$ )	IFOV (m)	Swath (km)	Dynamic Range (bits)
Panchromatic	0.45–0.90	1 $\times$ 1	11	11
Blue	0.45–0.53	4 $\times$ 4	11	11
Green	0.52–0.61	4 $\times$ 4	11	11
Red	0.64–0.72	4 $\times$ 4	11	11
Near IR	0.77–0.88	4 $\times$ 4	11	11

**A.3****Aircraft Scanners in the Visible and Infrared Regions****A.3.1****General Considerations**

Multispectral line scanners, similar in principle to the Landsat MSS and TM instruments, have been available for use in civil aircraft since the late 1960's and early 1970's. As with satellite image acquisition it is the forward motion of the aircraft that provides along track scanning whereas a rotating mirror or a linear detector array provides sensing in the across track direction.

There are several operational features that distinguish the data provided by aircraft scanners from that produced by satellite-borne devices. These are of significance to the image processing task. First, the data volume can be substantially higher. This is a result of having (i) a large number of spectral bands or channels available and (ii) a large number of pixels produced per mission, owing to the high spatial resolution available. Frequently up to 1000 pixels may be recorded across the swath, with many thousands of scan lines making up a flight line; each pixel is normally encoded to at least 8 bits.

A second feature of importance relates to field of view (FOV) – that is the scan angle either side of nadir over which data is recorded. This is depicted in Fig. A.4. In the case of aircraft scanning the FOV,  $2\gamma$ , is typically about 70 to 90°. Such a large angle is necessary to acquire an acceptable swath of data from aircraft altitudes. By comparison the FOV for the Landsats 1 to 3 is 11.56° while that for Landsats 4 and 5 is slightly larger at about 15°. The consequence of the larger FOV with aircraft



**Fig. A.4.** The concept of field of view (FOV) and instantaneous field of view (IFOV)

scanning is that significant distortions in image geometry can occur at the edges of the scan. Often these have to be corrected by digital processing.

Finally, the attitude stability of an aircraft as a remote sensing platform is much poorer than the stability of a satellite in orbit, particularly the Landsat 4 generation for which the pointing accuracy is  $0.01^\circ$  with a stability of  $10^{-6}$  degrees per second. Because of atmospheric turbulence, variations in aircraft attitude described by pitch, roll and yaw can lead to excessive image distortion. Sometimes the aircraft scanner is mounted on a three axis stabilized platform to minimise these variations. It is more common however to have the scanner fixed with respect to the aircraft body and utilize a variable sampling window on the data stream to compensate for aircraft roll.

Use of airborne multispectral scanners offers a number of benefits. Often the user can select the wavebands of interest in a particular application, and small bandwidths can be used. Also, the mission can be flown to specific user requirements concerning time of day, bearing angle and spatial resolution, the last being established by the aircraft height above ground level. As against these however, data acquisition from aircraft platforms is expensive by comparison with satellite recording since aircraft missions are generally flown for a single user and do not benefit from the volume market and synoptic view available to satellite data.

### A.3.2 Airborne Imaging Spectrometers

Since the mid 1980's the availability of new detector technologies has made possible the development of aircraft scanners capable of recording image data in a large number, typically hundreds, of spectral channels. For a given pixel, enough samples of its reflectance properties may be obtained by these instruments to allow very accurate characterisation of the pixel's spectral reflectance curve over the visible and reflected infrared region. Because of the large number of channels the data sets are often referred to as hyperspectral. These devices were the forerunners of instruments such as Hyperion, treated in Sect. A.2.12.

A good discussion of the development of imaging spectrometry may be found in Goetz et al. (1985) and Vane and Goetz (1988). Much of the work on those devices led to the development of similar spaceborne instruments.

Table A.17 summarises the characteristics of a selection of current aircraft imaging spectrometers.

## A.4 Spaceborne Imaging Radar Systems

### A.4.1

#### The Seasat SAR

The first earth observational space mission to carry a synthetic aperture imaging radar was the Seasat satellite launched in June 1978. Although only short lived it recorded about 126 million square kilometres of image data, including multiple coverage of many regions. Several other remote sensing instruments were also carried, including a radar altimeter, a scatterometer, a microwave radiometer and a visible and infrared imaging radiometer. Relevant characteristics of the satellite and its SAR are summarised in Table A.18. Polarization referred to in this table relates to the orientation of the electric field vector in the transmitted and received waveforms. Free space propagation of electromagnetic energy, such as that used for radar, takes place as a wave with electric and magnetic field vectors normal to each other and also normal to the direction of propagation. Should the electric field vector be parallel to the earth's surface, the wave is said to be horizontally polarized. Should it be vertical then the wave is said to be vertically polarized. A wavefront with a combination of the two will be either elliptically or circularly polarized. Even though one particular polarization might be adopted for transmission, some rotation can occur when the energy is reflected from the ground. Consequently at the receiver often both vertically and horizontally polarized components are available, each having its own diagnostic properties concerning the earth cover type being sensed. Whether one or the other, or both, are received depends upon the antenna used with the radar. In the case of Seasat, horizontally polarized radiation was transmitted ( $H$ ) and horizontally polarized returns were received ( $H$ ).

Further details on the Seasat SAR will be found in Elachi et al. (1982).

### A.4.2

#### Spaceborne (Shuttle) Imaging Radar-A (SIR-A)

A modified version of the Seasat SAR was flown as the SIR-A sensor on the second flight of Space Shuttle in November of 1981. Although the mission was shortened to three days, image data of about 10 million square kilometres was recorded. In contrast to Seasat however, in which the final image data was available digitally, the data in SIR-A was recorded and processed optically and thus is available only in film format. For digital processing therefore it is necessary to have areas of interest

**Table A.17.** Imaging spectrometers

Instrument	Spectral range μm	Spectral resolution nm	Dynamic range bits	IFOV mrad	Pixels per line
CASI-2 (Itres Research)	0.4 – 1 (288 channels)	2.2	12	1.3	512
CASI-3 (Itres Research)	0.4 – 1.05 (288 channels)	2.2	14	1.3	1490
DAIS 7915 (Geophysical Environmental Research Corp.)	0.4 – 1.0 (32 channels) 1.5 – 1.8 (8 channels) 2 – 2.5 (32 channels) 3 – 5 (1 channel) 8 – 12.6 (6 channels)	15 – 30 45 20 2000 900	15	3.3	512
AVIRIS (Airborne Visible and Infrared Imaging Spectrometer – JPL)	0.4 – 0.72 (31 channels) 0.69 – 1.30 (63 channels) 1.25 – 1.87 (63 channels) 1.84 – 2.45 (63 channels)	9.7 9.6 8.8 11.6	12	1	550
MIVIS (Daedalus Enterprises Inc.)	0.433 - 0.833 (20 channels) 1.15 – 1.55 (8 channels) 2.00 – 2.50 (64 channels) 8.20 – 12.70 (10 channels)	20 50 $\leq 500$	12	2	765
HYDICE (Hyperspectral Digital Image Collection Experiment US Naval Research Labs)	0.4 – 2.5 (206 channels)	7.6 – 14.9	12	0.5	320
HYMAP (Integrated Spectronics Pty Ltd)	0.44 – 0.88 0.881 – 1.335 1.4 – 1.81 1.95 – 2.5 (128 bands total)	16 13 12 16	12	$2.5 \times 2.0$	512

**Table A.18.** Characteristics of Seasat SAR, SIR-A, SIR-B, and SIR-C

	Seasat	SIR-A	SIR-B	SIR-C/X-SAR
Altitude	800 km	245 km	225–235 km	225 km
Wavelength	0.235 m	0.235 m	0.235 m	L – 0.235 m, C – 0.058 m, X – 0.031 m
Polarization	HH	HH	HH	HH, HV, VH, VV (only VV at X)
Incidence angle	20°	47°	15–57°	20–55°
Swath width	100 km	50 km	20–50 km	15–90 km
Range resolution	25 m	40 m	58–17 m	13–26 m (L, C), 10–20 (X)
Azimuth resolution	25 m	40 m	25 m	30 m

digitized from film using a device such as a scanning microdensitometer. A summary of SIR-A characteristics is given in Table A.18, wherein it will be seen that the incidence angle was chosen quite different from that for the Seasat SAR. Interesting features of landform can be brought out by processing the two together.

More details on SIR-A will be found in Elachi et al. (1982) and Elachi (1983).

### A.4.3

#### Spaceborne (Shuttle) Imaging Radar-B (SIR-B)

SIR-B, the second instrument in the NASA shuttle imaging radar program was carried on Space Shuttle mission 41 G in October 1984. Again the instrument was essentially the same as that used on Seasat and SIR-A, however the antenna was made mechanically steerable so that the incidence angle could be varied during the mission. Also about half the data was recorded digitally with the remainder being optically recorded. Details of the SIR-B mission are summarised in Table A.18; NASA (1984) contains further information on the instrument and experiments planned for the mission. Because of the variable incidence angle both the range resolution and swath width also varied accordingly.

### A.4.4

#### Spaceborne (Shuttle) Imaging Radar-C (SIR-C)/X-Band Synthetic Aperture Radar (X-SAR)

SIR-C/X-SAR, the third Shuttle radar mission, was carried out over two 10 day flights in April and September 1994. The SAR carried was the result of cooperation between NASA and DARA, the German Aerospace Agency, and had the characteristics indicated in Table A.18. Further details of the mission and the SAR will be found in Stofan et al. (1995) and Jordan et al. (1995).

### A.4.5

#### ERS-1,2

The European Remote Sensing Satellites ERS-1 and ERS-2 were launched in July 1991 and April 1995 respectively; they carry a number of sensors, one of which is a

**Table A.19.** Characteristics of free flying satellite SAR systems

	ERS-1, 2	JERS-1	Radarsat
Altitude	785 km	568 km	793–821 km
Wavelength*	0.057 m	0.235 m	0.057 m
Polarization	VV	HH	HH
Incidence angle	23°	35°	19–49°
Swath width	100 km	75 km	50–500 km
Range resolution	30 m	18 m	27 m
Azimuth resolution	30 m	18 m	19–24 m

\* 5.3 GHz for ERS 1, 2 and Radarsat and 1.28 Ghz for JERS-1

synthetic aperture radar intended largely for sea state and oceanographic applications. Characteristics of the radar are summarised in Table A.19.

#### **A.4.6 JERS-1**

The Japanese Earth Resources Satellite JERS-1 was launched in February 1992. It carries two imaging instruments; one is an optical sensor and the other an imaging radar. Table A.19 shows the design characteristics for the radar. The optical sensor, called OPS, has 8 wavebands between 0.52 µm and 2.40 µm with a swath width of 75 km and a dynamic range of 6 bits. The optical pixel size is 18.3 m (across track) × 24.2 m (along track). Provision is included for stereoscopic imaging.

#### **A.4.7 Radarsat**

Canada's Radarsat was launched on 4 November 1995; its SAR is able to operate in the standard and six non-standard modes, one of which will give a 518 km swath (Raney et al., 1991). Table A.19 lists the characteristics of the Radarsat SAR. Radarsat-2, scheduled for launch in 2005, will have an ultra-fine beam mode with 3 m resolution, and further polarisation options (VV, VH, HV).

#### **A.4.8 Shuttle Radar Topography Mission (SRTM)**

By deploying an outboard radar antenna 60 m from the space shuttle, along with the main antenna in the cargo bay, two simultaneous images can be obtained of the same region, but from different perspectives. Because of the coherent nature of the data, the two images can be interfered to reveal topographic detail of the earth's surface.

The Shuttle Radar Topography Mission in February 2000 used the SIR-C (C band)/X-SAR system to acquire interferometric data from which approximately 80% of the earth's land mass was imaged with 16 m absolute height accuracy and 20 m horizontal accuracy.

**A.4.9****Envisat Advanced Synthetic Aperture Radar (ASAR)**

The ASAR is an advanced version of the synthetic aperture radar from the ERS-1 and 2 missions. It operates at 5.331 GHz and incorporates a number of imaging modes that provide a variety of resolutions, polarisations and swath widths. Generally, the swath width is 100 km with the exception of wave mode (5 km) and wide swath width and global monitoring (400 km) products.

**A.4.10****The Advanced Land Observing Satellite (ALOS) PALSAR**

ALOS is scheduled for launch in 2005, and is designed as a follow on to JERS-1 and ADEOS (Midori). Besides PRISM (for stereoscopic mapping) and an AVNIR (see Sect. A.2.7) ALOS will carry a phased array L band SAR, to be known as PALSAR. The SAR will have a swath width of 70 km and a 2 look spatial resolution of 10 m in its observation mode, and a swath width of 250–360 km with a spatial resolution of 100 m in a scansar (wide swath width) mode.

**A.5****Aircraft Imaging Radar Systems**

Airborne imaging radar systems in SLAR and SAR technologies are also available. As with airborne multispectral scanners these offer a number of advantages over equivalent satellite based systems including flexibility in establishing mission parameters (bearing, incidence angle, spatial resolution etc.) and proprietary rights to data. However the cost of data acquisition is also high.

Table A.20 summarises the characteristics of three aircraft imaging radars, chosen to illustrate the operating parameters of these devices by comparison to satellite based systems. Note the band: wavelength designations – X: 0.030 m, C: 0.057 m, L: 0.235 m, P: 0.667 m. Note also that interferometric operation is also possible with the systems listed.

**Table A.20.** Representative aircraft synthetic aperture radar systems

	CCRS <sup>1</sup> SAR	DLR <sup>2</sup> ESAR	JPL <sup>3</sup> AIRSAR
Wavebands	X, C bands	X, C, L & P bands	C, L & P bands
Polarisation	HH, VV	multipolarisation	multipolarisation
Range resolution	6 m, 20 m	1.5 m	10 m
Azimuth resolution	6 m, 10 m	4–12 m	1 m
Interferometry	yes	yes	yes

<sup>1</sup> Canada Centre for Remote Sensing<sup>2</sup> Deutsche Forschungsanstalt für Luft- und Raumfahrt<sup>3</sup> Jet Propulsion Laboratory

## References for Appendix A

- C. Elachi (Chairman), 1983: Spaceborne Imaging Radar Symposium, Jet Propulsion Laboratory, January 17–20, JPL Publication, 83–11.
- C. Elachi, T. Bickell, R.L. Jordan and C. Wu, 1982: Spaceborne Synthetic Aperture Imaging Radars. Applications, Techniques and Technology. Proc. IEEE, 70, 1174–1209.
- NASA, 1984: The SIR-B Science Investigations Plan, Jet Propulsion Laboratory Publication, 84–3.
- A.F.H. Goetz, G.Vane, T.E. Solomon and B.N. Rock, 1985: Imaging Spectrometry for Earth Remote Sensing, Science, 228, 1147–1153.
- R.L. Jordan, B.L. Huneycutt and M. Werner, 1995: The SIR-C/X-SAR Synthetic Aperture Radar System. IEEE Trans. Geoscience and Remote Sensing, 33, 829–839.
- R.K. Raney, A.P. Luscombe, E.J. Lanham and S. Ahmed, 1991: Radarsat. Proc. IEEE, 79, 839–849.
- E.R. Stofen, D.L. Evans, C. Schmullius, B. Holt, J.J. Plaut, J. van Zyl, S.D. Wall and J. Way, 1995: An Overview of Results of Spaceborne Imaging Radar-C, X-Band Synthetic Aperture Radar (SIR-C/X-SAR). IEEE Trans. Geoscience and Remote Sensing, 33, 817–828.
- G. Vane and A.F.H. Goetz, 1988: Terrestrial Imaging Spectroscopy, Remote Sensing of Environment, 21, 311–332.

## Appendix B

### Satellite Altitudes and Periods

Civilian remote sensing satellites are generally launched into circular orbits. By equating centripetal acceleration in a circular orbit with the acceleration of gravity it can be shown (Duck and King, 1983) that the orbital period corresponding to an orbital radius  $r$  is given by

$$T = 2\pi \sqrt{r^3/\mu} \quad (\text{B.1})$$

where  $\mu$  is the earth gravitational constant, with a value of  $3.986 \times 10^{14} \text{ m}^3 \text{s}^{-2}$ . The corresponding orbital angular velocity is

$$\dot{\theta} = \sqrt{\mu/r^3} \text{ rad} \cdot \text{s}^{-1} \quad (\text{B.2})$$

The orbital radius  $r$  can be written as the sum of the earth radius  $r_e$  and the altitude of a satellite above the earth,  $h$ :

$$r = r_e + h \quad (\text{B.3})$$

where  $r_e = 6.378 \text{ Mm}$ . Thus the effective velocity of a satellite over the ground (at its sub-nadir point) *ignoring earth rotation*, is given by

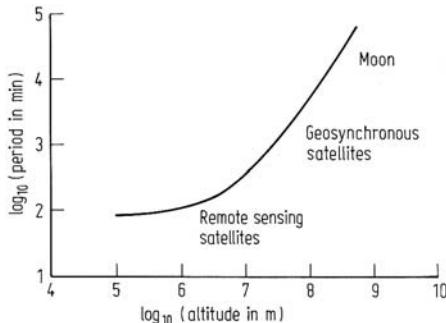
$$v = r_e \dot{\theta} = r_e \sqrt{\mu/(r_e + h)^3} \quad (\text{B.4})$$

The actual velocity over the earth's surface taking into account the earth's rotation depends upon the orbit's inclination (measured as an angle  $i$  anticlockwise from the equator on an ascending pass – i.e. at the so-called ascending node) and the latitude at which the velocity is of interest.

Let the earth rotational velocity at the equator be  $v_e$  (to the east). Then at latitude  $\phi$ , the surface velocity of a point on the earth will be  $v_e \cos \phi \cos i$ . Therefore the actual ground track velocity of a satellite at altitude  $h$  and orbital inclination  $i$  is given by

$$v_s = r_e \sqrt{\mu/(r_e + h)^3} \pm v_e \cos \phi \cos i \quad (\text{B.5})$$

where the + sign applies when the component of earth rotation opposes the satellite motion (i.e. on descending nodes for inclinations less than  $90^\circ$  or for ascending nodes with inclinations greater than  $90^\circ$ ). Otherwise the negative sign is used.



**Fig. B.1.** Satellite periods versus altitude above the earth's surface, for circular orbits

Equations (B.1) to (B.5) can be used to derive some numbers of significance for remote sensing missions, although it is to be stressed here that the equations apply only for circular orbits and a spherical earth.

Figure B.1 shows a plot of orbital period (in minutes) as a function of satellite altitude plotted on logarithmic coordinates. This has been derived from (B.1) and (B.3). Some significant altitudes to note are (i)  $h = 907$  km, at which  $T = 103$  min; being the approximate period of the first three Landsat satellites, (ii)  $h = 35,800$  km at which  $T = 24$  hours, being the so-called geosynchronous orbit – if this is established over the equator then the satellite appears stationary to a point on the ground; this is the orbit used by many communication satellites, (iii)  $h = 380$  Mm at which  $T = 28$  days – this is the orbit of the moon.

Consider now a calculation of the time taken for Landsat 1 to acquire a 185 km frame of MSS data. This can be found by determining the local velocity. For the Landsat satellite the orbital inclination is  $100^\circ$ ; at Sydney Australia the latitude is  $34^\circ\text{S}$ . From (B.5) this gives

$$v_s = 6.392 \text{ km s}^{-1}.$$

Therefore 185 km requires 28.9 s to record.

## References for Appendix B

K.I. Duck and J.C. King, 1983: Orbital Mechanics for Remote Sensing. In: R.N. Colwell (Ed.). Manual of Remote Sensing, 2e, American Society of Photogrammetry, Falls Church.

## Appendix C

# Binary Representation of Decimal Numbers

In digital data handling we frequently refer to numbers in binary form; this is because computers and their associated storage media represent data in this format. In the binary system the numbers are arranged in columns that represent powers of 2 while in the decimal system numbers are arranged in columns that are powers of 10. Thus whereas we can count up to 9 in each column in the decimal system we can only count up to one in each binary column. From the right, the columns represent  $2^0$ ,  $2^1$ ,  $2^2$  etc., so that the decimal numbers between 0 and 7 have the binary versions:

Decimal	Binary		
	$2^2$	$2^1$	$2^0$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1 (i.e. $2 + 1$ )
4	1	0	0
5	1	0	1 (i.e. $4 + 1$ )
6	1	1	0 (i.e. $4 + 2$ )
7	1	1	1 (i.e. $4 + 2 + 1$ )

The digits in the binary system are referred to as bits. In the above example it can be seen that by using just 3 binary digits it is not possible to represent decimal numbers beyond 7 – i.e. a total of 8 decimal numbers altogether, including 0. To represent 16 decimal numbers, which could be 16 levels of brightness in remote sensing image data between 0 and 15, it is necessary to have a binary “word” with 4 bits. In that case the word 1111 is equivalent to decimal 15. In this way it is readily shown that the numbers of decimal values that can be represented by various numbers of binary digits are:

Number of bits	Number of decimal levels
1	2 (i.e. 0,1)
2	4 (0, 1, 2, 3,)
3	8 (0, ..., 7)
4	16 (0, ..., 15)
5	32 etc.
6	64
7	128
8	256
9	512
10	1024
11	2048
12	4096

An eight bit word, which can represent 256 decimal numbers between 0 and 255, is referred to as a *byte* and is a fundamental data unit used in computers.

# Appendix D

## Essential Results from Vector and Matrix Algebra

### D.1

#### Definition of a Vector and a Matrix

The pixels in an image can be plotted in a rectangular co-ordinate system according to their brightness values in each band. For example, for Landsat MSS bands 5 and 7 a vegetation pixel would appear somewhat as shown in Fig. D.1. The pixel can be described by its co-ordinates (10, 40); this will be a set of four numbers if all 4 MSS bands are considered, in which case the co-ordinate system also is four dimensional. For the rest of this discussion only two dimensions will be used for illustration but the results apply to any number. For example a 7 dimensional space would be required for Landsat ETM+ data. The vector space will have several hundred dimensions for imaging spectrometer data.

An alternative but *equivalent* means by which the pixel point can be represented is as a *vector* drawn from the origin, as illustrated in Fig. D.2. In this context the vector is simply an arrow that points to the pixel. While we never actually draw the vector as such it is useful to remember that it is implied in much of what follows.

Mathematically a vector from the origin is described in the following way. First we define so-called *unit vectors* along the co-ordinate directions. These are simply direction indicators, which for the two dimensional case are as shown in Fig. D.3. With these, the vector is written as

$$\mathbf{x} = x_1 \hat{i} + x_2 \hat{j}$$

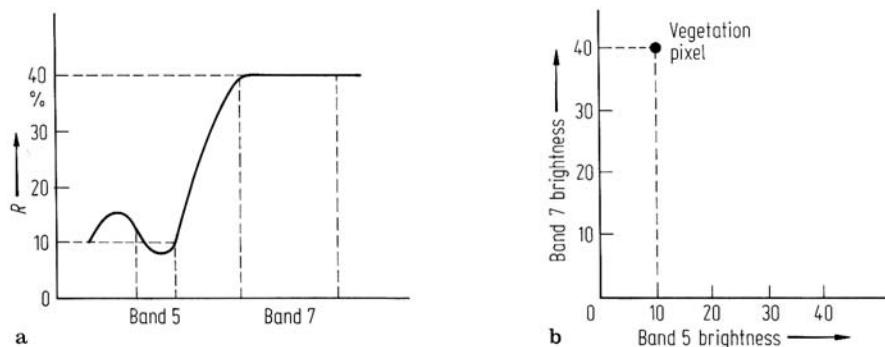
Symbol used to denote vector

Co-ordinate horizontally

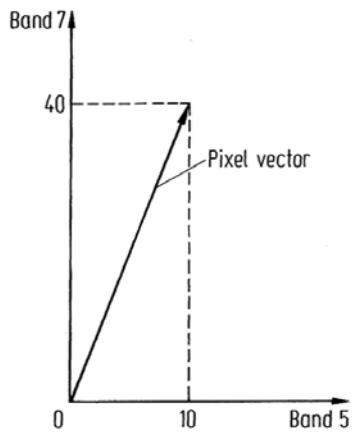
Co-ordinate Vertically

Unit vector in the horizontal ( $x_1$ ) direction

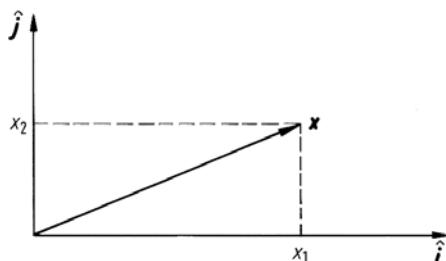
Unit vector in the vertical ( $x_2$ ) direction



**Fig. D.1.** a Spectral reflectance characteristic of vegetation; b typical vegetation pixel plotted in a rectangular co-ordinate system



**Fig. D.2.** Representation of a pixel point in multi-spectral space by a vector drawn from the origin



**Fig. D.3.** Definition of unit vectors

In “shorthand” form we represent the vector as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

which is properly referred to as a column vector, owing to its vertical arrangement. Note that the unit vectors, and thus the corresponding co-ordinate directions are

implied by the ordering in the column. Sometimes a row version is used. This is called the *transpose* of  $\mathbf{x}$  and is written as

$$\mathbf{x}^t = [x_1 \quad x_2].$$

Recall that for Landsat MSS data  $\mathbf{x}$  will have 4 column entries representing the four response values for the pixel that  $\mathbf{x}$  describes.

Sometimes we might wish to create another vector  $\mathbf{y}$  from an existing vector  $\mathbf{x}$ . For illustration, if we take both to be just two dimensional then the components of  $\mathbf{y}$  can be obtained most generally according to the pair of equations

$$\begin{aligned} y_1 &= m_{11}x_1 + m_{12}x_2 \\ y_2 &= m_{21}x_1 + m_{22}x_2 \end{aligned}$$

i.e. the components of  $\mathbf{y}$  are just (linear) combinations of those of  $\mathbf{x}$ . In shorthand this *transformation* of the vector is expressed as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

or

$$\mathbf{y} = M\mathbf{x}$$

where  $M$  is referred to as a *matrix* of coefficients. By comparing the previous expressions, note how a multiplication of a matrix by a vector is carried out.

## D.2 Properties of Matrices

The *inverse* of  $M$  is called  $M^{-1}$  and is defined by

$$MM^{-1} = I$$

where  $I$  is the *identity matrix*

$$\begin{bmatrix} 1 & 0 & & \\ 0 & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

which, if used to transform the vector  $\mathbf{x}$ , will leave it unchanged. This can be seen if it is used in place of  $M$  in the equations above. The inverse of a matrix is not always easily computed. It should be noted however that it can be expressed as

$$M^{-1} = M^*/|M|$$

where  $M^*$  is called the *adjoint* of  $M$  and  $|M|$  is called its *determinant*. The adjoint, in theory, is a *transposed matrix of cofactors*. This is not important in general for remote sensing since all the calculations are usually performed with software that includes

a procedure for inverting a matrix. However it is useful for illustration purposes to know that the adjoint for a  $2 \times 2$  matrix is:

$$M^* = \begin{bmatrix} m_{22} & -m_{12} \\ -m_{21} & m_{11} \end{bmatrix} \quad \text{when } M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

Similarly large order determinant calculations are carried out by computer. It is only necessary to note, for illustration, that

$$\begin{vmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{vmatrix} = m_{11}m_{22} - m_{21}m_{12} = \text{a scalar constant.}$$

### D.3

## Multiplication, Addition and Subtraction of Matrices

If  $M$  and  $N$  are two matrices, chosen as  $2 \times 2$  for illustration and defined as

$$M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \quad N = \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix}$$

then

$$M \pm N = \begin{bmatrix} m_{11} \pm n_{11} & m_{12} \pm n_{12} \\ m_{21} \pm n_{21} & m_{22} \pm n_{22} \end{bmatrix}$$

and

$$\begin{aligned} MN &= \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \times \begin{bmatrix} n_{11} & n_{12} \\ n_{21} & n_{22} \end{bmatrix} \\ &= \begin{bmatrix} m_{11}n_{11} + m_{12}n_{21} & m_{11}n_{12} + m_{12}n_{22} \\ m_{21}n_{11} + m_{22}n_{21} & m_{21}n_{12} + m_{22}n_{22} \end{bmatrix} \end{aligned}$$

Note that the last expression is obtained by multiplying, term by term, the rows of the first matrix by the columns of the second. Within each multiplication the terms are summed. This pattern holds for larger matrices.

Division is not defined as a matrix operation. Rather its place is taken by the definition of a matrix inverse, as in the above section.

### D.4

## The Eigenvalues and Eigenvectors of a Matrix

We have discussed the matrix  $M$  above as a matrix that transforms one vector to another, (alternatively it can be used to transform the co-ordinate system in which a point or vector is described). It is relevant at this stage to ask if there is a vector that can be multiplied by a simple (but in general complex) number and thus be transformed in exactly the same manner as it would be had it been multiplied by the matrix  $M$ . In other words can we find a vector  $x$  in our co-ordinate space and a (complex) number  $\lambda$  such that

$M\mathbf{x} = \lambda\mathbf{x}$  (i.e.  $\mathbf{y} = \lambda\mathbf{x}$  is equivalent to  $\mathbf{y} = M\mathbf{x}$ ).

This implies

$$M\mathbf{x} - \lambda\mathbf{x} = 0$$

or

$$(M - \lambda I)\mathbf{x} = 0 \quad (\text{D.1})$$

The theory of simultaneous equations tells us that for this equation to be true it is necessary to have either  $\mathbf{x} = 0$  or

$$|M - \lambda I| = 0 \quad (\text{D.2})$$

This expression is a polynomial equation in  $\lambda$ . When evaluated it yields values for  $\lambda$ . When these are substituted into (D.1) the vectors  $\mathbf{x}$  corresponding to those  $\lambda$  will be found. Those  $\lambda$ 's are called the *eigenvalues* of  $M$  and the associated  $\mathbf{x}$ 's are called the *eigenvectors*.

## D.5

### Some Important Matrix, Vector Operations

If  $\mathbf{x}$  is a column vector, say  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$  then

$$\begin{aligned} \mathbf{x}\mathbf{x}^t &= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} [x_1 \quad x_2] \\ &= \begin{bmatrix} x_1^2 & x_1x_2 \\ x_1x_2 & x_2^2 \end{bmatrix} \end{aligned}$$

i.e. a matrix

and

$$\begin{aligned} \mathbf{x}^t\mathbf{x} &= [x_1 \quad x_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ &= x_1^2 + x_2^2 \end{aligned}$$

i.e. a constant (this is often referred to as the dot product, scalar product or inner product).

## D.6

### An Orthogonal Matrix – The Concept of Matrix Transpose

The inverse of an *orthogonal* matrix is identical to its transpose. Thus, if  $M$  is orthogonal then

$$M^{-1} = M^t \quad \text{and} \quad M^t M = I,$$

where  $M^t$  is the transpose of the matrix, given by rotating all the elements about the principal diagonal (that which runs through the elements  $m_{11}, m_{22}, m_{33}, \dots$ ).

## D.7 Diagonalisation of a Matrix

Consider a transformation matrix  $M$  such that

$$\mathbf{y} = M\mathbf{x}.$$

As before the eigenvalues  $\lambda_i$  of  $M$  and their associated eigenvectors  $\mathbf{x}_i$  are defined by the expression

$$\lambda_i \mathbf{x}_i = M\mathbf{x}_i, \quad i = 1, \dots, n$$

where  $n$  is the number of distinct eigenvalues.

These  $n$  different equations can be expressed in the compact manner

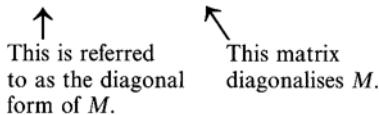
$$X\Lambda = MX$$

where  $\Lambda$  is the diagonal matrix

$$\begin{bmatrix} \lambda_1 & 0 \\ \ddots & \ddots \\ 0 & \lambda_n \end{bmatrix}$$

and  $X$  is the matrix of eigenvectors ( $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$ ).

Consequently  $\Lambda = X^{-1} M X$


 This diagram illustrates the decomposition of a matrix  $M$  into three components:  $X$ ,  $\Lambda$ , and  $X^{-1}$ . An arrow points from  $M$  to  $X$ , labeled "This matrix diagonalises  $M$ ". Another arrow points from  $M$  to  $\Lambda$ , labeled "This is referred to as the diagonal form of  $M$ ".

If  $X$  is orthogonal then  $X^{-1} = X^t$  so that  $\Lambda = X^t M X$ .

# Appendix E

## Some Fundamental Material from Probability and Statistics

### E.1 Conditional Probability

It is the purpose of this Appendix to outline some of the fundamental statistical concepts commonly used in remote sensing theoretical developments. Remote sensing terminology is used throughout and an emphasis is placed on understanding rather than theoretical rigour.

The expression  $p(x)$  is interpreted as the probability that the event  $x$  occurs. In the case of remote sensing, if  $x$  is a pixel vector,  $p(x)$  is the probability that a pixel can be found at position  $x$  in multispectral space.

Often we wish to know the probability of an event occurring conditional upon some other event or circumstance. This is written as  $p(x|y)$  which is expressed as the probability that  $x$  occurs given that  $y$  is specified. As an illustration  $p(x|\omega_i)$  is the probability of finding a pixel at position  $x$  in multispectral space, given that we are interested in class  $\omega_i$  – i.e. it is the probability that a pixel from class  $\omega_i$  exists at position  $x$ . These  $p(x|y)$  are referred to as *conditional probabilities*; the available  $y$  generally form a complete set. In the case of remote sensing the set of  $\omega_i$ ,  $i = 1, \dots, M$  are the complete set of spectral classes used to describe the image data for a particular exercise. If we know the complete set of  $p(x|\omega_i)$  – which are often referred to as the *class conditional probabilities* – then we can determine  $p(x)$  in the following manner. Consider the product  $p(x|\omega_i)p(\omega_i)$  where  $p(\omega_i)$  is the probability that class  $\omega_i$  occurs in the image (or that a pixel selected at random will come from class  $\omega_i$ ). The product is the probability that a pixel at position  $x$  in multispectral space is an  $\omega_i$  pixel. The probability that a pixel from *any* class can be found at position  $x$  clearly is the sum of the probabilities that pixels will be found there from all the available classes. In other words

$$p(x) = \sum_{i=1}^M p(x|\omega_i)p(\omega_i) \quad (\text{E.1})$$

The product  $p(\mathbf{x}|\omega_i)p(\omega_i)$  is called the *joint probability* of the “events”  $\mathbf{x}$  and  $\omega_i$ . It is interpreted strictly as the probability that a pixel occurs at position  $\mathbf{x}$  and that the class is  $\omega_i$  (this is different from the probability that a pixel occurs at position  $\mathbf{x}$  given that we are interested in class  $\omega_i$ ). The joint probability is written

$$p(\mathbf{x}, \omega_i) = p(\mathbf{x}|\omega_i) p(\omega_i) \quad (\text{E.2a})$$

We can also write

$$p(\omega_i, \mathbf{x}) = p(\omega_i|\mathbf{x}) p(\mathbf{x}) \quad (\text{E.2b})$$

where  $p(\omega_i|\mathbf{x})$  is the conditional probability that expresses the likelihood that the class is  $\omega_i$  given that we are examining a pixel at position  $\mathbf{x}$  in multispectral space. Often this is called the *posterior* probability of class  $\omega_i$ . Again  $p(\omega_i, \mathbf{x})$  is the probability that  $\omega_i$  and  $\mathbf{x}$  exist together, which is the same as  $p(\mathbf{x}, \omega_i)$ . As a consequence, from (E.2a) and (E.2b)

$$p(\omega_i|\mathbf{x}) = p(\mathbf{x}|\omega_i) p(\omega_i)/p(\mathbf{x}) \quad (\text{E.3})$$

which is known as Bayes’ theorem (Freund, 1992).

## E.2

### The Normal Probability Distribution

#### E.2.1

#### The Univariate Case

The class conditional probabilities  $p(\mathbf{x}|\omega_i)$  in remote sensing are frequently assumed to belong to a normal probability distribution. In the case of a one dimensional spectral space this is described by

$$p(x|\omega_i) = (2\pi)^{-1/2} \sigma_i^{-1} \exp \left\{ -\frac{1}{2}(x - m_i)^2/\sigma_i^2 \right\} \quad (\text{E.4})$$

in which  $x$  is the single spectral variable,  $m_i$  is the mean value of  $x$  and  $\sigma_i$  is its standard deviation; the square of the standard deviation,  $\sigma_i^2$ , is called the variance of the distribution. The mean is referred to also as the expected value of  $x$  since, on the average, it is the value of  $x$  that will be observed on many trials. It is computed as the mean value of a large number of samples of  $x$ . The variance of the normal distribution is found as the expected value of the difference squared of  $x$  from its mean. A simple average of this squared difference gives a biased estimate. An unbiased estimate is obtained from (Freund, 1992)

$$\sigma_i^2 = \frac{1}{q_i - 1} \sum_{j=1}^{q_i} (x_j - m_i)^2 \quad (\text{E.5})$$

where  $q_i$  is the number of pixels in class  $\omega_i$  and  $x_j$  is the  $j$ th sample.

### E.2.2

#### The Multivariate Case

The one dimensional case just outlined is seldom encountered in remote sensing, but it serves as a basis for inducing the nature of the multivariate normal probability distribution, without the need for theoretical development. Several texts treat the bivariate case – i.e. that where  $\mathbf{x}$  is two dimensional – and these could be consulted should a simple multivariate case be of interest without vector and matrix notation (Nilsson, 1965 and 1990; Swain and Davis, 1978; Freund, 1992).

Consider (E.4) and see how it can be modified to accommodate a multidimensional  $\mathbf{x}$ . First, and logically,  $x$  must be replaced by  $\mathbf{x}$ . Likewise the univariate mean  $m_i$  must be replaced by its multivariate counterpart  $\mathbf{m}_i$ . The variance  $\sigma_i^2$  in (E.4) must be modified, not only to take account of multidimensionality but also to include the effect of correlation between spectral bands. This role is filled by the covariance matrix  $\Sigma_i$  defined by

$$\Sigma_i = \mathcal{E} \{ (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^t \} \quad (\text{E-6a})$$

where  $\mathcal{E}$  is the expectation operator and the superscript “ $t$ ” is the vector transpose operation. An unbiased estimate for  $\Sigma_i$  is given by

$$\Sigma_i = \frac{1}{q_i - 1} \sum_{j=1}^{q_i} \{ (\mathbf{x}_j - \mathbf{m}_i)(\mathbf{x}_j - \mathbf{m}_i)^t \} \quad (\text{E.6b})$$

Inside the exponent in (E.4) the variance  $\sigma_i^2$  appears in the denominator. In its multivariate extension the covariance matrix is inverted and inserted into the numerator of the exponent. Moreover the squared difference between  $\mathbf{x}$  and  $\mathbf{m}_i$  is expressed using the vector transpose expression  $(\mathbf{x} - \mathbf{m}_i)^t(\mathbf{x} - \mathbf{m}_i)$ . Together these allow the exponent to be recast as  $-\frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1}(\mathbf{x} - \mathbf{m}_i)$ . We now turn our attention to the pre-exponential term. First we need to obtain a multivariate form for the reciprocal of the standard deviation. This is achieved first by using the determinant of the covariance matrix as a measure of its size (see Appendix D) – giving a single number measure of variance – and then taking its square root. Finally the term  $(2\pi)^{-1/2}$  needs to be replaced by  $(2\pi)^{-N/2}$ , leading to the complete form of the multivariate normal distribution for  $N$  spectral dimensions

$$p(\mathbf{x} | \omega_i) = (2\pi)^{-N/2} |\Sigma_i|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^t \Sigma_i^{-1}(\mathbf{x} - \mathbf{m}_i) \right\} \quad (\text{E.7})$$

## References for Appendix E

Along with vector and matrix analysis and calculus, a sound understanding of probability and statistics is important in developing a high degree of skill in quantitative remote sensing. This is necessary not only to appreciate algorithm development but also because of the role of statistical sampling techniques and the like when dealing with sampled data. The depth of treatment in this appendix and in the body of the book has been sufficient for a first level

appreciation of quantitative methods. The reader wishing to develop further understanding, particularly of important concepts in probability and statistics, would be well advised to consult well-known standard treatments such as Freund (1992) and Feller (1967).

- W. Feller, 1967: An Introduction to Probability Theory and its Applications, 2e, N.Y., Wiley.  
J.E. Freund, 1992: Mathematical Statistics, 5e, Englewood Cliffs, N.J., Prentice Hall.  
N.J. Nilsson, 1965: Learning Machines, N.Y., McGraw-Hill.  
N.J. Nilsson, 1990: Mathematical Foundations of Learning Machines. Palo Alto, Morgan Kaufmann.  
P.H. Swain and S.M. Davis (Eds.), 1978: Remote Sensing: The Quantitative Approach, N.Y., McGraw-Hill.

# Appendix F

## Penalty Function Derivation

### of the Maximum Likelihood Decision Rule

#### F.1

#### Loss Functions and Conditional Average Loss

The derivation of maximum likelihood classification in Sect. 8.2 is generally acceptable for remote sensing applications and is used widely. However it is based implicitly on the understanding that misclassifying any particular pixel is no more significant than misclassifying any other pixel in an image. The more general approach presented in the following allows the user to specify the importance of making certain labelling errors compared with others. For example, for crop classification involving two sub-classes of wheat it would probably be less of a problem if a particular wheat pixel was erroneously classified into the other sub-class than it would if it were classified as water.

To develop the general method we introduce the penalty function, or loss function

$$\lambda(i|k) \quad i, k = 1, \dots, M \quad (\text{F.1})$$

This is a measure of the loss or penalty incurred when an algorithm erroneously labels a pixel as belonging to class  $\omega_i$  when in reality the pixel is from class  $\omega_k$ . It is reasonable to expect that  $\lambda(i|i) = 0$  for all  $i$ ; this implies there is no penalty for a correct classification. In principle, there are  $M^2$  distinct values of  $\lambda(i|k)$  where  $M$  is the number of classes.

The penalty incurred by erroneously labelling a pixel at position  $\mathbf{x}$  in multispectral space into class  $\omega_i$  is

$$\lambda(i|k) p(\omega_k|\mathbf{x})$$

where the pixel comes correctly from class  $\omega_k$  and  $p(\omega_k|\mathbf{x})$  is the posterior probability that  $\omega_k$  is the correct class for pixels at  $\mathbf{x}$ . Averaging this over all possible  $\omega_k$  we have the average loss, correctly referred to as the *conditional average loss*, associated with labelling a pixel as belonging to class  $\omega_i$ . This is given by

$$L_{\mathbf{x}}(\omega_i) = \sum_{k=1}^M \lambda(i|k) p(\omega_k|\mathbf{x}) \quad (\text{F.2})$$

and is a measure of the accumulated penalty incurred given the pixel could have belonged to any of the available classes and that we have available the penalty functions relating all the classes to class  $\omega_i$ . Clearly, a useful decision rule for assigning a label to a pixel is to choose that class for which the conditional average loss is the smallest, viz

$$\mathbf{x} \in \omega_i \quad \text{if} \quad L_{\mathbf{x}}(\omega_i) < L_{\mathbf{x}}(\omega_j) \quad \text{for all } j \neq i. \quad (\text{F.3})$$

An algorithm that implements (F.3) is often referred to as a Bayes' optimal algorithm.

Even if the  $\lambda(i|k)$  were known, the  $p(\omega_k|\mathbf{x})$  usually are not. Therefore, as in Sect. 8.2.2 we adopt Bayes' theorem which allows the posterior probabilities to be expressed in terms of the class probability distribution functions  $p(\mathbf{x}|\omega_k)$ ; viz

$$p(\omega_k|\mathbf{x}) = p(\mathbf{x}|\omega_k)p(\omega_k)/p(\mathbf{x})$$

where  $p(\omega_k)$  is the class prior probability. Using this in (F.2) gives

$$L_{\mathbf{x}}(\omega_i) = \frac{1}{p(\mathbf{x})} l_{\mathbf{x}}(\omega_i)$$

with

$$l_{\mathbf{x}}(\omega_i) = \sum_{k=1}^M \lambda(i|k) p(\mathbf{x}|\omega_k) p(\omega_k). \quad (\text{F.4})$$

Since  $p(\mathbf{x})$  is common to all classes it is sufficient to decide class membership on the basis of the  $l_{\mathbf{x}}(\omega_i)$ .

## F.2 A Particular Loss Function

Suppose  $\lambda(i|k) = 1 - \Phi_{ik}$  with  $\Phi_{ii} = 1$  and  $\Phi_{ik}$  ( $k \neq i$ ) to be defined. Then (F.4) can be expressed

$$\begin{aligned} l_{\mathbf{x}}(\omega_i) &= \sum_{k=1}^M p(\mathbf{x}|\omega_k) p(\omega_k) - \sum_{k=1}^M \Phi_{ik} p(\mathbf{x}|\omega_k) p(\omega_k) \\ &= p(\mathbf{x}) - g_i(\mathbf{x}) \end{aligned}$$

with

$$g_i(\mathbf{x}) = \sum_{k=1}^M \Phi_{ik} p(\mathbf{x}|\omega_k) p(\omega_k) \quad (\text{F.5})$$

Again since  $p(\mathbf{x})$  is common to all classes it does not aid discrimination and thus can be removed from the conditional average loss expression, leaving just  $l_{\mathbf{x}}(\omega_i) = -g_i(\mathbf{x})$ . Because of the minus sign in this expression we can then decide the "least cost" labelling of a pattern at position  $\mathbf{x}$  in multispectral space according to maximisation of the *discriminant junction*  $g_i(\mathbf{x})$ , viz

$$\mathbf{x} \in \omega_i \quad \text{if} \quad g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all} \quad j \neq i. \quad (\text{F.6})$$

It is of interest at this stage to put

$\Phi_{ik} = \delta_{ik}$ , the Kroneker delta function,

defined by

$$\begin{aligned}\delta_{ik} &= 1 \quad \text{for } i = k \\ &= 0 \quad \text{for } i \neq k.\end{aligned}$$

Equation (F.5) then becomes

$$g_i(\mathbf{x}) = p(\mathbf{x}|\omega_i) p(\omega_i)$$

so that the decision rule in (F.6) is

$$\mathbf{x} \in \omega_i \quad \text{if} \quad p(\mathbf{x}|\omega_i) p(\omega_i) > p(\mathbf{x}|\omega_j) p(\omega_j) \quad \text{for all} \quad j \neq i$$

which is the classification rule adopted in (8.3) in Chap. 8. Frequently this is referred to as the unconditional maximum likelihood decision rule.

## References for Appendix F

Nilsson (1965) gives an excellent account of the derivation of the maximum likelihood decision rule based upon penalty functions in the manner just derived. Duda, Hart and Stork (2001) also cover the topic in detail.

R.O. Duda, P.E. Hart and R.G. Stork, 2001: Pattern Classification, N.Y., Wiley.  
N.J. Nilsson, 1965: Learning Machines, N.Y., McGraw-Hill.

# Subject Index

- Absorption features 360, 366, 369, 370
- Abundance maps 385
- Accuracy
  - classification 303
  - cross validation 307
  - leave one out method 307
  - producer's 304
  - user's 304
- Activation function 232
- Active remote sensing 3
- Adjoint – see Matrix
- Advanced Earth Observing Satellite (ADEOS) 398
- Advanced Land Imager (ALI) 401, 403
- Advanced Land Observing Satellite (ALOS)
  - 411
- Advanced Spaceborne Thermal Emission and Reflection Spectrometer (ASTER)
  - 401, 404
- Advanced Synthetic Aperture Radar (ASAR)
  - 411
- Advanced TIROS-N 389
- Advanced Very High Resolution Radiometer (AVHRR) 390
- Advanced Visible and Near Infrared Radio-meter (AVNIR) 398, 399
- Advanced Wide Field Sensor (AWiFS) 401
- Agglomerative hierarchical clustering – see
  - Clustering
- Airborne imaging spectrometers 406
- Airborne Visible and Infrared Spectrometer (AVIRIS) 5, 13, 362, 408
- Aircraft scanners 405
- Airsar 412
- Aliasing 64, 174, 192
- Allocation – see Classification
- Among categories variance 280
- Among class covariance matrix 280
- Aperiodic functions 169
- Aqua 401, 404
- Aspect ratio distortion 44, 54
- Atmosphere 27, 28, 31, 33, 34, 360, 366
- Atmospheric absorption 29, 360
  - water vapour 29, 360, 361, 366
- Atmospheric and solar curve models – see
  - Radiometric correction
- Atmospheric scattering 29, 360, 361, 366
- Atmospheric turbulence 43
- ATREM – see Radiometric correction
- Automatic contrast enhancement – see
  - Contrast enhancement
- Azimuth 13
- Azimuth resolution 14, 409, 410
- Backpropagation 234
- Band arithmetic 137, 160, 292
- Band ratios 160, 292
- Band-to-band errors 28
- Bathymetry 102, 104
- Bayes' classification – see Classification
- Bayes' theorem 195, 216, 424
- Bhattacharyya distance 273, 379, 381
- Binary decision tree – see Decision tree classifier
- Binary number system 415
- Binary spectral codes 371

- Binomial probability distribution 305
- Bispectral plot 302, 311, 320
- Bit shuffling – see Fast Fourier transform
- Black body radiation 6
- Block-based maximum likelihood classification 375
- Block diagonal matrix – see Matrix
- Box car filtering – see Filtering
- Byte 416
- Calibration – see Radiometric correction
- Canonical analysis 279
  - segmented 380
- CASI 408
- Categorization – see Classification
- Change detection 160, 314
- Characteristic equation 142
- Charge coupled device (CCD) 10, 397
- Chi-squared distribution 197
- Class 75
- Class conditional probability 218, 423
- Class signature 194, 295
- Classification 72
  - accuracy 303
  - Bayes' 194, 428
  - committee 335
  - consensus theoretic 336
  - context 209
  - cost 206, 267
  - decision tree 225, 231, 243, 321
  - effect of resampling 298
  - hierarchical 321
  - k nearest neighbour 207
  - layered 231
  - Mahalanobis 206
  - maximum likelihood 194, 195, 427
  - minimum distance 201
  - multistage 321
  - non-parametric 80, 219
  - parallelepiped 204
  - progressive two-class decision 324
  - supervised 78, 193, 295
  - table look up 207
  - unsupervised 78, 249, 255, 299
- Classification methodologies
  - hybrid 301
  - supervised 295
  - unsupervised 299
- Cluster maps 255
- Clustering 78, 249
  - agglomerative hierarchical 260
  - divisive hierarchical 261
  - histogram peak selection 263
  - isodata 251
  - iterative optimization 251
  - migrating means 251
  - single pass 257
- Clustering cost 254
- Clustering criteria 249
- Clusters
  - deletion 252
  - merging 252
  - splitting 254
- Coastal Zone Colour Scanner (CZCS) 389
- Code book 383
- Cofactor – see Matrix
- Colour composite image product 69, 380
- Colour density slicing 104
- Colour infrared 69
- Compatibility coefficient 213
- Complex exponential function 166
- Complex permittivity 7
- Compression of hyperspectral data 382
- Conditional average loss 427
- Conditional probability 423
- Confusion matrix 166, 303
- Conjugate gradient 235
- Consensus theoretic classification – see Classification
- Context classification – see Classification
- Contingency table – see Confusion matrix
- Contrast enhancement 83
  - automatic 88
  - exponential 89
  - Gaussian 99
  - linear 86
  - logarithmic 89
  - multicycle 104
  - piecewise linear 89
  - saturating linear 88
  - Taylor method 151
- Contrast matching
  - image-to-image 98
  - image-to-mathematical reference 99
- Control points – see Ground control points
- Convolution 110, 127, 171
  - graphical 172

- integral 111, 171
- spatial frequency domain 189
- theorem 173, 188
- time domain 111
- two dimensional 111, 188
- two dimensional discrete 111, 189
- Corner reflector effect 7
- Correction increment 223
- Correlation matrix 71, 139, 363, 375
- Covariance matrix 80, 138, 196, 375, 381
- Crabbing 64
- Critical distance 258
- Cross validation – see Accuracy
- Cumulative histogram 90
  
- DAIS Scanner 408
- Dark current 32
- Data base management system (DBMS) 18
- Data fusion 333
- Decision boundaries 79
- Decision boundary feature extraction – see Feature selection
- Decision rule
  - Bayes' optimal 428
  - evidential 341
  - k nearest neighbour 208
  - maximum likelihood 195
  - minimum distance 202
  - multisource 335
  - unconditional maximum likelihood 429
- Decision surfaces 196, 204
- Decision tree classifier 321
  - decision tree design 323
  - binary decision tree 225, 324
  - error accumulation 327
- Dendrogram 261
- Density slicing
  - black and white 101
  - colour 104
- Destriping 37
- Detector gain 32
- Detector offset 32
- Determinant – see Matrix
- Diffuse reflection 8
- Dirac delta function 166
  - sifting property 168
- Discrete cosine transform (DCT) 383
- Discrete Fourier transform 176
  - computation 179
- inverse 177
- of an image 184
- properties 178
- Discriminant analysis 282
  - nonparametric 286
- Discriminant analysis feature selection 285
- Discriminant function
  - k nearest neighbour 208
  - Markov random field 219
  - maximum likelihood 196
  - minimum distance 202
  - support vector machine 230
  - unconditional maximum likelihood 428
- Divergence 269
  - average 271
  - of a pair of normal distributions 270
  - transformed 274
  - use for feature selection 271
- Divisive hierarchical clustering – see Clustering
- Dynamic range 3
  
- Earth curvature 27, 37, 42
- Earth gravitational constant 413
- Earth radius 38, 413
- Earth rotation effect on image geometry 27, 37, 38
- Edge detection 118
- Edge enhancement 118
  - by subtractive smoothing 123
- Eigenvalue – see Matrix
- Eigenvector – see Matrix
- Emittance 6
- Empirical Line – see Radiometric correction
- End member 385
- Endorsement 351
- Energy 129
- Enhanced Thematic Mapper (ETM) 393, 396
- Enhanced Thematic Mapper+ (ETM+) 393, 396
- Entropy 129
- ENVI – see Software systems
- Envisat 411
- EO-1 401, 403
- ER Mapper – see Software systems
- Error correction feedback 223
- Error matrix 303

- Errors
  - commission 303
  - geometric 27, 37
  - omission 303
  - radiometric 27
- ERS-1,2 409, 410
- Euclidean distance 202, 250
- Evidence 338
- Evidential interval 339
- Evidential mass 338
- Exclusive OR operator 373
- Exponential contrast enhancement – see Contrast enhancement
- Extraction and Classification of Homogeneous Objects (ECHO) 210
- Fast Fourier transform 179
  - bit shuffling 184
  - computational cost 183
- Feature reduction 268, 377
  - by transformation 276
- Feature selection 271, 301, 378
  - by canonical analysis 279
  - by principal components 277, 381
  - Decision boundary feature extraction 286
  - Non-parametric discriminant analysis 286
  - Non-parametric weighted feature extraction 290
- Field classification 294
- Field of view (FOV) 11
- Filtering
  - box car 116
  - convolution 127
  - high pass 188
  - mean value 115
  - median 116
  - modal 134
  - low pass 115, 188
  - smoothing 115, 188
  - spatial frequency domain 187
- Finite sensor scan rate 43, 77
- Fisher criterion 285
- Flat Field – see Radiometric correction
- Fourier series 168
- Fourier transformation 113, 165, 169
  - discrete – see Discrete Fourier transform
  - fast – see Fast Fourier transform
- Fraunhofer absorption 360
- Gaussian contrast enhancement – see Contrast enhancement
- Gaussian mixture models 208
- Gaussian probability distribution – see Normal probability distribution
- Generalized eigenvalue equation 282
- Geocoding 56
- Geographic information system (GIS) 18, 333
- Geometric correction 46
- Geometric enhancement 72, 109, 187
- Georeferencing 56
- Geostationary Meteorological Satellite (GMS) 391
- Geostationary Operational Environmental Satellite (GOES) 391
- Geosynchronous
  - orbit 414
  - weather satellites 391
- Global Imager (GLI) 399
- Grey level co-occurrence matrix (GLCM) 128
- Ground control points 47
  - choice 51, 58
- Ground range resolution 14
- Hamming distance 373
- Haze removal 35
- Heaviside (unit) step function 168
- Hierarchical classification – see Classification
- Hierarchical clustering – see Clustering
- High Resolution Geometry (HRG) sensor 397, 398
- High Resolution Spectroscopy (HRS) sensor 397
- High Resolution Visible (HRV) sensor 397, 398
- High Resolution Visible and Infrared (HRVIR) sensor 397, 398
- Histogram 84
- Histogram equalization 90
  - anomalous 95
- Histogram matching 97
- Histogram modification 84
- Hotelling transformation – see Principal components transformation
- Hughes phenomenon 328, 364, 375

- Hybrid classification – see Classification methodologies
- HYDICE 408
- HYMAP 408
- Hyperion 401, 403
- Hyperspectral 10, 68, 328, 360
  - spectral profiles 360
  - spectral redundancy 362
- Ikonos 405
- Image arithmetic – see Band arithmetic
- Image filtering – see Filtering
- Image histogram – see Histogram
- Image interpretation 67
- Image orientation to north-south 55
- Image rotation 61
- Image to image contrast matching – see Contrast matching
- Image to image registration – see Registration
- Image to map registration – see Registration
- Imaging spectrometers 10, 359
- Improved Limb Atmospheric Sensor (ILAS) 398
- Improved Tiros Operational Satellite (ITOS) 389
- Impulse function – see Dirac delta function
- Impulse response 111
- Impulsive noise 118
- Indian Remote Sensing Satellite (IRS) 401
- Inference mechanism 345, 346
- Information class 75
- Instantaneous field of view (IFOV) 10, 11, 395
- Interferometric Monitor for Greenhouse Gases (IMG) 398
- Interpoint (L1) distance 250
- Interpolation 48, 65
  - bilinear 48
  - cubic convolution 50
  - nearest neighbour 48
- Interpolative zoom 61
- Irradiance 28
- Isodata clustering algorithm – see Clustering
- Iterative optimization clustering – see Clustering
- Jeffries-Matusita distance 273
  - average 274
- of a pair of Normal distributions 273
- JERS-1 410
- Joint probability 424
- Justifier 350
- k nearest neighbour classifier – see Classification
- Kappa coefficient 304
- Karhunen-Loeve transformation – see Principal components transformation
- Kauth-Thomas – see Tasseled cap Transformation
- Kernels 230
- Knowledge-based image analysis 342
- Kronecker delta function 219, 429
- Labelling – see Classification
- Landsat 391
- Leave one out method – see Accuracy
- Layered classification – see Classification
- Likelihood ratio 269
- Line detection 125
- Line striping 32, 36
- Linear contrast enhancement – see Contrast enhancement
- Linear detector array (CCD) 10
- Linear discrimination 202, 220, 226
- Linear Imaging Self Scanner (LISS) 401
  - LISS I 402
  - LISS II 402
  - LISS III 402
  - LISS IV 402
- Linear system 110
- Linear system theory 110
- Log residuals – see Radiometric correction
- Logarithmic contrast enhancement – see Contrast enhancement
- Look up tables 72, 83
- Loss function – see Penalty function
- Lowtran 7 – see Radiometric correction
- Mahalanobis – see Classification
- Mahalanobis distance 207
- Map accuracy – see Accuracy, Classification accuracy
- Maple – see Software systems
- Mapping polynomials 42
- Marine Observation Satellite (MOS) 399, 400

- Markov random fields 216
- Mathematica – see Software systems
- Mathematical modelling of geometric errors 54
- MATLAB – see Software systems
- Matrix 419
- addition of 420
  - adjoint 419
  - block diagonal 375
  - cofactor 419
  - determinant of 419
  - diagonal 381, 422
  - diagonalization of 422
  - eigenvalues of 142, 420
  - eigenvectors of 142, 420
  - identity 419
  - inverse 419
  - multiplication of 420
  - orthogonal 421
  - pseudo inverse 386
  - subtraction of 420
  - symmetric 140
  - trace of 143
  - transpose of 419, 421
- Maximum likelihood classification – see Classification
- Mean value smoothing – see Filtering
- Mean vector 80, 138, 196
- Median filtering – see Filtering
- Midori 399
- Mie (aerosol) scattering 30
- Migrating means clustering – see Clustering
- Minimum distance classification – see Classification
- MIVIS 408
- Mixed data types – see Spatial data sources
- Mixed pixels 302, 385
- Modal filtering – see Filtering
- Moderate Resolution Imaging Spectrometer (MODIS) 401, 404
- Mosaicing 97, 99
- MSU-SK 401, 402
- Multicycle contrast enhancement – see Contrast enhancement
- Multilayer perceptron 232
- MultiSpec – see Software systems
- Multispectral line scanner 10
- Multispectral Scanner (MSS) 393, 394
- Multispectral space 75
- Multispectrum Electronic Self Scanning Radiometer (MESSR) 400
- Multistage classification – see Classification
- NASA Spectrometer (NSCAT) 398
- Neighbourhood function 212
- Neighbourhood operations 109, 190
- Neural networks 80, 232, 296
- Nimbus satellites 389
- NOAA satellites 389
- Noise fraction 155
- Non-parametric classification – see Classification
- Non-parametric discriminant analysis – see Feature selection
- Non-parametric weighted feature extraction – see Feature selection
- Normal probability distribution
- multivariate 80, 196, 425
  - univariate 424
- Nugget variance 132
- Nyquist rate 174
- Ocean Colour and Temperature Sensor (OCTS) 398, 399
- Ocean Colour Monitor (OCM) 401
- Opinion Pools
- linear 336
  - logarithmic 337
- OPS 410
- Optical thickness 33
- OrbView-2 satellite 399
- Orthogonal sum 340
- PALSAR 411
- Pan (on IRS) 402
- Panoramic distortion 37, 39, 55
- Parallelepiped classification – see Classification
- Passive remote sensing 3
- Path radiance 30, 44
- Pattern hyperplane 222
- Penalty function 427
- Periodic functions 168
- Photointerpretation 2, 67
- Piecewise linear contrast modification – see Contrast enhancement
- Pitch 43
- Pixel replicative zoom 61

- Pixel spectra 10, 68
- Pixel vector 77, 417
- Platform altitude variations 37, 43
- Platform attitude variations 37, 43
- Platform ephemeris 43
- Platform velocity variations 37, 43
- Plausibility 339
- Point operations 83
- Point spread function 111, 209
- Polar orbit 390
- Polarization (electromagnetic) 9, 407
- Polarization and Directionality of the Earth's Reflectance (POLDER) 398
- Posterior probability 195, 424, 427
- Prewitt operator 122
- Principal components transformation 137
  - for change detection 314
  - for data reduction 154, 277, 381
  - noise adjusted 154
  - segmented 379
- Prior probability 195, 428
- Probabilistic label relaxation 211
- Processing element 232
- Progressive two-class decision classifier – see Classification
- Pseudocolouring 104
- Push broom scanner 10
- Pyramid images 19, 24
- Qualitative reasoning 342
- Quantitative analysis 67, 72, 193
- Radsat 410
- Radiance 29
- Radiometric correction 32, 366
  - 5S Code 366
  - ATREM 366
  - empirical line 368
  - flat field 368
  - haze removal 35
  - log residuals 367
  - Lowtran 7 366
  - Modtran 3 366
- Radiometric distortion 27
- Radiometric enhancement – see Contrast enhancement
- Radiometric modification – see Contrast enhancement
- Radiometric resolution – see Resolution
- Range resolution 14, 409, 410
- Raster format 17
- Rayleigh scattering 30
- Rectification – see Geometric correction
- Reflectance 29, 360, 366
  - apparent 366
  - real 366
  - scaled 366
- Registration 27, 56
  - image to image 57
  - to map grid 51
- Regularised covariance estimates 381
- Repeat cycle 392
- Resampling 48
  - effect on classification 298
- Resolution
  - radiometric 3
  - spatial 3
- RESURS-01 401, 403
- Retroreflector in Space (RIS) 399
- Return Beam Vidicon (RBV) 393
- Roberts operator 121
- Roll 43
- Rule-based image processing 133
- Sampling theorem 174
- Sampling theory 173
- Satellite
  - altitude 413
  - orbit 414
- Saturating linear contrast enhancement – see Contrast enhancement
- S-bend distortion 41
- Scalar image 83
- Scale (image) 21
- Scale changing 61
- Scan time skew distortion 43
- Scanning Multichannel Microwave Radio meter (SMMR) 389
- Scattering
  - aerosol 30
  - corner reflector 7
  - diffuse 7
  - Mie 30
  - Rayleigh 30
  - specular 7
  - surface 7
  - volume 7
- Scattering coefficient 6

- Scattering matrix
  - between class 287
  - within class 288
- Seasat 407
- SeaStar 399, 400
- Sea-Viewing Wide Field of View Sensor (SeaWiFS) 399, 400
- Semivariogram 131
- Sensor non-linearities 37, 45
- Separability 267
  - maximum likelihood classification 267, 268
  - minimum distance classification 276
- Sequential similarity detection algorithm (SSDA) 57
- Shape detection and recognition 132
- Shape factor 132
- Sharpening 123
- Shuttle Imaging Radar (Spaceborne Imaging Radar)
  - SIR-A 407
  - SIR-B 409
  - SIR-C/X-SAR 409
- Shuttle Radar Topography Mission (SRTM) 410
- Side looking airborne radar (SLAR) 12
- Signature 194, 295
- Sill 132
- Similarity metrics 249
- Single pass clustering algorithm – see Clustering
- Sky irradiance 30, 360
- Slant range 14
- Slant range resolution 14
- Smoothing – see Filtering
- Sobel operator 122
- Software systems
  - ENVI 203
  - ER Mapper 203
  - Maple 145
  - MATLAB 145, 242
  - Mathematica 145
  - MultiSpec 203, 210, 259, 275
- Solar spectrum 7, 360, 366, 379
- Spatial context 209
- Spatial data sources 15
- Spatial derivative 121
- Spatial frequency 185
- Spatial resolution – see Resolution
- Speckle 9
- Spectral Angle Mapping (SAM) 368
- Spectral class 75, 249, 302
- Spectral irradiance 29
- Spectral library 369, 371
- Spectral reflectance characteristics
  - soil 5
  - vegetation 5
  - water 5
- Spectral slope 373
- Spectral unmixing 385
- Specular reflection 7
- SPOT satellite 397
- Stacked vector 385
- Standard deviation multiplier 258
- Stereoscopic viewing 397
- Stratified random sampling 305
- Strip generation parameter 259
- Sum of squared errors measure (SSE) 250
- Sun synchronous 9
- Supervised classification – see Classification
- Supervised relaxation labelling 337
- Support 339
- Support vector 229
- Support vector classifier 226
- Support vector machine 80, 226, 296
- Symmetric matrix – see Matrix
- Synthetic aperture radar (SAR) 12
- System function 111
- Table look up classification – see Classification
- Tasseled cap transformation 156
- Taylor method of contrast enhancement – see Contrast enhancement
- Templates 109
  - edge detection 120
  - line detecting 125
  - non-linear 125
  - semi-linear 125
  - smoothing 115
- Terra 401, 404
- Texture 128
- Thematic Mapper (TM) 393, 396
- Theory of Evidence 338
- Thermal infrared 6
- Threshold logic unit 224

- Thresholds
  - in image smoothing 115
  - in maximum likelihood classification 197
  - in minimum distance classification 204
- TIROS 389
- TIROS Operational Satellite (TOS) 389
- Total Ozone Mapping Spectrometer (TOMS)
  - 398
- Trace – see Matrix
- Tracking and Data Relay Satellite (TDRS)
  - 392
- Training data 80, 193, 296
- Training fields 193, 295
- Training pixels, number required 199, 364, 375
- Transfer characteristics of radiation detectors
  - 32
- Transformed divergence 274
- Transformed vegetation index 292
- Transmittance (atmospheric) 30
- Transpose
  - of a matrix – see Matrix
  - of a vector 419
- Uniform histogram 90
- Unit step waveform – see Heaviside step function
- Unsupervised classification – see Classification
- Vector
  - pixel 77, 138, 417
  - transpose 419
  - unit 417
- Vector format 17
- Vector gradient 121
- Vector image 83
- Vector quantisation 383
- Vector to raster conversion 17
- Vegetation index 160, 292
- Vegetation instrument 397, 398
- Visible and Infrared Spin Scan Radiometer (VISSR) 391
- Visible and Thermal Infrared Radiometer (VTIR) 400
- Wavelet transform 190
- Weather satellites 389
- Weight point 222
- Weight space 222
- Weight vector 220
- Wide Field Sensor (WiFS) 401
- Window functions 190
- Within categories variance 280
- Within class covariance matrix 280
- Yaw 43
- Zooming 61
  - interpolative 61
  - pixel replicative 61