The University of Aston in Birmingham

PETROGRAPHIC IMAGE ANALYSIS AND UNDERSTANDING

Robert Clark Thomson

Doctor of Philosophy

1991

Summary

This study considers the application of image analysis in petrography and investigates the possibilities for advancing existing techniques by introducing feature extraction and analysis capabilities of a higher level than those currently employed. The aim is to construct relevant, useful descriptions of crystal form and inter-crystal relations in polycrystalline igneous rock sections. Such descriptions cannot be derived until the 'ownership' of boundaries between adjacent crystals has been established: this is the fundamental problem of crystal boundary assignment. An analysis of this problem establishes key image features which reveal boundary ownership; a set of explicit analysis rules is presented.

A petrographic image analysis scheme based on these principles is outlined and the implementation of key components of the scheme considered. An algorithm for the extraction and symbolic representation of image structural information is developed. A new multiscale analysis algorithm which produces a hierarchical description of the linear and near-linear structure on a contour is presented in detail. Novel techniques for symmetry analysis are developed. The analyses considered contribute both to the solution of the boundary assignment problem and to the construction of geologically useful descriptions of crystal form.

The analysis scheme which is developed employs grouping principles such as collinearity, parallelism, symmetry and continuity, so providing a link between this study and more general work in perceptual grouping and intermediate level computer vision. Consequently, the techniques developed in this study may be expected to find wider application beyond the petrographic domain.

Key Words: Image analysis, image understanding, igneous petrography, shape analysis, symmetry analysis

ACKNOWLEDGEMENTS

I should like to express my thanks to my supervisor, Dr Ela Claridge, for her advice, encouragement and support.

Financial support provided by the University of Aston in Birmingham is acknowledged

CONTENTS

Page

Summary 2
Acknowledgements 3
List of Figures 10
List of Tables 14

Chapter

1. Introduction 15
1.1 Image processing, analysis and understanding 15
1.2 Thesis overview 16
1.2.1 Thesis structure 17

2. Image Processing in Petrography 19
2.1 Introduction 19
2.2 Petrography 19
2.2.1 Petrology 19
2.2.2 Rock micrographs 20
2.2.3 The mineralogical microscope 21
2.2.4 Crystal optics 22
2.2.5 Features of mineral sections in transmitted light 23
2.2.6 Reflected light microscopy 25
2.3 Image processing in petrography: a review 26
2.3.1 Shape analysis 27
2.3.2 Mathematical morphology 29
2.3.2.1 Erosion and dilation 30
2.3.2.2 Pore complex analysis 31
2.3.3 Quantitative image analysis 32
2.3.3.1 Applications 32
2.3.3.2 Example image analysis systems 35
2.3.4 Approaches to image segmentation 37
2.3.4.1 Non-optical approaches 43
2.3.5 Summary of findings 44
2.4 Image understanding in petrography: a proposal 44

3. Crystalline Fabric Analysis....47

3.1 Introduction....47

3.2 Igneous rocks....48

3.2.1 Igneous petrogenesis....48

3.2.2 Description of crystal form in igneous rocks....50

3.2.2.1 Quality of development of faces....51

3.2.2.2 Crystal habit....52

3.2.3 Crystal shape: convexity and non-convexity....53

3.3 Metamorphic rocks....54

3.3.1 Metamorphic petrogenesis....54

3.3.2 Description of crystal form in metamorphic rocks....54

3.3.3 Metamorphic textures c.f. igneous textures....56

3.4 Crystal symmetry description....56

3.5 The identification of characteristic boundaries....58

3.5.1 Amstutz and Giger's quantitative fabric analysis....59

3.5.2 Establishing crystal 'precedence'....61

3.5.2.1 Crystal inclusion and convexity relations....61

3.5.2.2 False corners....63

3.5.2.3 Symmetry....64

3.5.3 Accommodating non-convexity....65

3.5.4 Precedence indicators: summary of rules....69

3.5.4.1 Rules in combination....70

3.6 Implications for subsequent work....71

4. A Spatial Information Database....74

4.1 Introduction....74

4.1.1 Digital connectivity....74

4.2 Image spatial information and its representation....76

4.2.1 Spatial information from crystal profiles images....76

4.2.2 Representing spatial information: some examples....77

4.2.2.1 The 'segmented image database'....77

4.2.2.2 The 'patchery data structure'....78

4.2.2.3 A network data structure....79

4.2.2.4 The McGill relational database....81

4.2.3 The structure of the SPIN spatial information database....82

4.3 The extraction of image spatial information....85

4.3.1 Structural feature extraction: some examples....86

4.3.1.1 Structural analysis in GIAPP 87
4.3.1.2 A graph traversal-based scheme 88
4.3.2 Constructing SPIN: the Spinster algorithm 93
4.3.2.1 Outer and inner region borders 93
4.3.2.2 The boundary tracking algorithm 94
4.3.2.3 Spinster: implementation details 96
4.3.2.4 SPIN database example 98
4.3.2.5 Future work: coping with cracks 101
4.4 Conclusions 103

5. Multiscale Contour Analysis 104
5.1 Introduction 104
5.1.1 Problems in crystal profile analysis 104
5.1.2 Chapter outline 106
5.2 Contour analysis: a review 107
5.2.1 Fixed resolution polygonal approximation of contours 107
5.2.2 Multiple resolution contour analysis 109
5.2.2.1 Scale-space 109
5.2.2.2 Davis' angle and side detection 111
5.2.2.3 Curve partitioning 113
5.2.2.4 Lowe's algorithms 115
5.2.2.5 The work of Bengtson and Eklundh 118
5.3 A new algorithm for multiscale contour analysis 121
5.3.1 Creating an intermediate representation for contours 122
5.3.2 An outline of the intermediate representation 125
5.3.2.1 'cpspace' 125
5.3.2.2 'mmspace' 126
5.3.3 Analysis of the intermediate representation 129
5.3.3.1 Choice of arc lengths 133
5.3.3.2 Finding stable zones in mmspace 135
5.3.3.3 Choice of thresholds 136
5.3.3.4 Corner detection 138
5.3.3.5 Illustrative examples: finding possible faces 139
5.3.4 Combining structural interpretations 149
5.3.4.1 Preliminary filter 150
5.3.4.2 Elimination of conflicts and redundancies 151
5.3.4.3 Enforcing a hierarchy of faces 153
5.3.4.4 Supplementary filter: pruning the face tree 157

5.3.4.5 Illustrative examples: refining the face sets 158
5.3.5 Generating limited error polygonal approximations 166
5.3.6 Development of the algorithm 168
5.3.6.1 History of development 169
5.3.6.2 Current form of the algorithm 171
5.3.7 Detail of the algorithm 172
5.3.7.1 Producing the lists of zone and face records 174
5.3.7.2 Finding face overlap and sorting face records 177
5.3.7.3 Greedy selection 179
5.3.7.4 Endpoint adjustment 180
5.4 Testing and evaluation 182
5.4.1 Metrics for performance 183
5.4.1.1 Performance analysis method 184
5.4.1.2 Establishing a reference solution 185
5.4.1.3 Example performance analysis 186
5.4.2 Range of tests and results 187
5.4.2.1 Topaz 188
5.4.2.2 Lamprophyllite 190
5.4.2.3 Olivine 1 191
5.4.2.4 Allanite 193
5.4.2.5 Augite 1 195
5.4.2.6 Augite 2 198
5.4.2.7 Nosean 199
5.4.2.8 Olivine 2 201
5.4.2.9 Arrow 203
5.4.2.10 Heptagon 205
5.4.3 Analysis of results 207
5.4.4 Computational aspects 209
5.5 Conclusions and further work 211
5.5.1 Further algorithm development 212
5.5.2 Summary and conclusions 214

6. Symmetry Analysis 216
6.1 Introduction 216
6.1.1 Database representation of faces and polygonal approximations 217
6.1.2 Pruning the face tree 219
6.2 Symmetry analysis: a review 223
6.3 The reduced stereogram 225

6.3.1 Intimations of symmetry 226
6.4 Representation of stereogram and symmetries 227
6.4.1 Significance measures 229
6.5 Building a stereogram 231
6.5.1 Conflicting interpretations 234
6.5.1.1 Generating maximal consistent sets of faces 234
6.5.1.2 Stereogram construction using the face tree 236
6.5.1.3 Building the master stereogram 237
6.6 Implementation of symmetry analyses 239
6.6.1 Analysis of simple face sets 239
6.6.1.1 Parallelisms 239
6.6.1.2 Mirror planes 239
6.6.2 Illustrative examples 240
6.6.2.1 Topaz crystal analysis 240
6.6.2.2 Nosean crystal analysis 243
6.6.3 Analysis of face trees 245
6.6.3.1 Parallelisms 247
6.6.3.2 Mirror planes 247
6.6.4 Illustrative examples 247
6.6.4.1 Topaz crystal analysis 248
6.6.4.2 Lamprophyllite crystal analysis 249
6.6.4.3 Olivine crystal analysis 254
6.6.4.4 Augite crystal analysis 257
6.7 Further development and conclusions 260
6.7.1 Further work 260
6.7.2 Summary and conclusions 262

7. System Overview 264
7.1 Introduction 264
7.2 Current implementation status 265
7.2.1 Building the SPIN database 265
7.2.2 Identifying possible faces 265
7.2.3 Convexity analysis 266
7.2.4 Cleavage analysis 267
7.2.5 Symmetry analysis 268
7.2.6 False corner analysis 268
7.3 System control: the supervisor 269
7.3.1 Control strategy 269

8. Summary and Conclusions.........271
8.1 Overview.........271
8.2 Thesis review.........272
8.3 Further developments.........275

References.........277

LIST OF FIGURES

Figure ____ Page

3.1 Synthetic section image 50
3.2 Euhedral and anhedral crystals 51
3.3 A crystal, its sketch stereogram, and symmetry analysis 58
3.4 'False corners' 60
3.5 Inclusion styles 62
3.6 Partial enclosure with a straight interface 62
3.7 Illustration of the limitations of the 'false corner' heuristic 63
3.8 Interfaces assigned via symmetry 64
3.9 Olivine-malignite micrograph image 66
3.10 Analysis of an interface using the convex hull 68
3.11 Petrographic image analysis system outline 73

4.1 Example of the digital connectivity paradox 75
4.2 Freeman chain coding 75
4.3 The patchery data structure 79
4.4 An inter-region border of variable thinning 91
4.5 Correct line tracking and vertex recognition 91
4.6 Boundary tracking start conditions 95
4.7 The four possible situations during tracking 95
4.8 A test image and the labelling used in the SPIN database 98
4.9 Border and internal cracks 101
4.10 A border crack on a boundary segment, and its removal 102
4.11 Tracking border cracks around holes 102

5.1 Contours showing structure at more than one resolution 105
5.2 Alternative polygonal approximations of a contour 105
5.3 A signal and its interval tree represented as a tessellation of scale-space 110
5.4 Example of a hierarchical curve structure and its graph 112
5.5 Strip segments in a strip tree. 116
5.6 A curve segment and its corresponding strip tree 116
5.7 A contour and two polygonal approximations created via strip trees 118
5.8 Sketch of 'mmspace' representation of an irregular quadrilateral 126

5.9 Sketch of 'mmspace' representation for a semi-regular polygon....................127
5.10 An irregular contour and a sketch of its mmspace representation....................130
5.11 Face rectangles and face markers in mmspace, with associated face tree............132
5.12 A corner within a 'stable zone'..138
5.13 'Pentagon' test contour and its mmspace representation..........................140
5.14 Face markers in mmspace found via three sets of thresholds, using the slower arc length progression...140
5.15 Faces corresponding to the markers of 5.14, and their superimposition............141
5.16 Faces found via three threshold sets and their superimposition, using the faster arc length progression...143
5.17 Topaz crystal profile and its mmspace representation..............................144
5.18 Face markers in mmspace found via three sets of thresholds, using the slower arc length progression...145
5.19 Faces corresponding to face markers of figure 5.26...............................145
5.20 Faces found via three threshold sets, faster arc length progression used...........146
5.21 Lamprophyllite crystal profile image...147
5.22 Faces found from three density slicing passes, with and without special category for perfectly straight faces...148
5.23 Contour segment and sketch of associated face markers in mmspace...............154
5.24 Contour section and sketch of associated face markers in mmspace................155
5.25 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour...158
5.26 Faces markers corresponding to output from greedy selection and output from endpoint adjustment, with mmspace..159
5.27 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour...160
5.28 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour...161
5.29 Faces markers corresponding to output from greedy selection and output from endpoint adjustment, with mmspace..162
5.30 Faces input to greedy selection, output from greedy selection, and output from endpoint adjustment...163
5.31 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour...163
5.32 Faces output from greedy selection and faces output from endpoint adjustment....164
5.33 Faces output from greedy selection, faces output from endpoint adjustment, faces following the supplementary filter, and the original contour........................165
5.34 Contour segment requiring double breakpoints.......................................167

5.35 Fixed error approximations to a contour..168
5.36 Instructions for preparing hand analyses 185
5.37 The consensus solution and multiscale analysis of the 'pentagon' test contour.....186
5.38 Topaz crystal profile 188
5.39 Analysis of test profile using slower and faster arc length progressions.............188
5.40 Analysis of test profile using Lowe's algorithm..189
5.41 Lamprophyllite crystal profile..190
5.42 Analysis of test profile using slower and faster arc length progressions.............190
5.43 Analysis of profile using Lowe's algorithm...191
5.44 First olivine crystal profile..191
5.45 Analysis of test profile using slower and faster arc length progressions.............192
5.46 Analysis of profile using Lowe's algorithm...192
5.47 Allanite crystal profile...193
5.48 Analysis of test profile using slower and faster arc length progressions.............193
5.49 Analysis of allanite profile using Lowe's algorithm..194
5.50 First version of first augite crystal profile 195
5.51 Analysis of test profile using slower and faster arc length progressions.............195
5.52 Analysis of test profile using Lowe's algorithm 196
5.53 Second version of first augite crystal profile 196
5.54 Analysis of test profile using slower and faster arc length progressions.............196
5.55 Analysis of second version of first augite profile using Lowe's algorithm..........197
5.56 Second augite crystal profile 198
5.57 Analysis of test profile using slower and faster arc length progressions.............198
5.58 Analysis of profile using Lowe's algorithm...199
5.59 Nosean crystal profile...199
5.60 Analysis of test profile using slower and faster arc length progressions.............200
5.61 Analysis of test profile using Lowe's algorithm 200
5.62 Second olivine crystal profile 201
5.63 Analysis of test profile using slower and faster arc length progressions.............202
5.64 Analysis of test profile using Lowe's algorithm 202
5.65 Arrow test profile at two scales..203
5.66 Analysis of test profile using slower and faster arc length progressions.............203
5.67 Analysis of smaller arrow profile using Lowe's algorithm.......................................204
5.68 Analysis of larger 'arrow' test profile using slower and faster arc length progressions 204
5.69 Heptagon test profile at two scales 205
5.70 Analysis of smaller 'heptagon' test profile using slower and faster arc length progressions...205

5.71 Analysis of smaller 'heptagon' profile using Lowe's algorithm 206
5.72 Analysis of larger 'heptagon' test profile using slower and faster arc length progressions 207
5.73 Time *v* contour length, for creation of intermediate representation 210
5.74 Timings for second phase of algorithm 211
5.75 Simplifying arc-chord computations 213

6.1 A face-tree of faces and the database representation 218
6.2 Topaz crystal profile, original face set, and pruned face set 220
6.3 Nosean crystal profile, original face set, and pruned face set 221
6.4 Olivine crystal profile, original face set, and pruned face set 221
6.5 Augite crystal profile, original face set, and pruned face set 222
6.6 Lamprophyllite crystal profile, original face set, and pruned face set 223
6.7 A crystal profile, its stereogram, and analysis 225
6.8 Supporting evidence for a mirror plane 227
6.9 A set of crystal faces and associated stereogram 233
6.10 Pruned face tree for topaz crystal profile 234
6.11 Generating maximal consistent face sets 235
6.12 Stereogram and the two major mirror planes found for topaz crystal 242
6.13 Faces found on nosean crystal profile 243
6.14 Symmetry analysis of nosean crystal 244
6.15 Faces found on topaz crystal profile, and their labels 248
6.16 Final selection of faces for topaz crystal 249
6.17 Hierarchy of faces for lamprophyllite crystal profile 250
6.18 Final selection of faces for lamprophyllite crystal 253
6.19 Hierarchy of faces for olivine crystal profile 254
6.20 Olivine crystal face set (d) and original profile 256
6.21 Hierarchy of faces for augite crystal profile 257
6.22 Symmetry analysis of augite crystal 259
6.23 Augite crystal face set (a) and original profile 260
6.24 Two attenuation functions 261

7.1 Petrographic image analysis system outline 264

LIST OF TABLES

Table ______ Page

3.1 Precedence analysis of figure 3.10 61

4.1 A SPIN database for figure 5.8 111

5.1 Performance summary for 'pentagon' example 187
5.2 Performance summary for topaz example 189
5.3 Performance summary for lamprophyllite example 191
5.4 Performance summary for first olivine example 192
5.5 Performance summary for allanite example 194
5.6 Performance summary for two versions of first augite example 197
5.7 Performance summary for second augite example 199
5.8 Performance summary for nosean example 201
5.9 Performance summary for second olivine example 202
5.10 Performance summary for 'arrow' example 204
5.11 Performance summary for 'heptagon' example 206
5.12 Performance summary for all real data examples 207
5.13 Performance summary for all test examples 207

6.1 The stereogram list for the face set of figure 6.17 233
6.2 Master stereogram representing figure 6.18 238
6.3 Results of mirror plane analysis on figure 6.9 241
6.4 The stereogram list for the face set of figure 6.13 243
6.5 Results of mirror plane analysis on figure 6.13 244
6.6 Symmetry analyses for 4 interpretations of figure 6.15 248
6.7 The stereogram list for the face set of figure 6.17 251
6.8 Symmetry analyses for 4 likeliest interpretations of figure 6.17 252
6.9 The stereogram list for the face set of figure 6.19 254
6.10 List of parallelisms derived from master stereogram list 255
6.11 Symmetry analyses for 4 interpretations of figure 6.19 255
6.12 The master stereogram list for the face-tree of figure 6.21 257
6.13 List of parallelisms derived from master stereogram list 258
6.14 Symmetry analyses for 3 interpretations of figure 6.20 258

7.1 Multiscale analysis performance summary for ten test examples 266

CHAPTER 1

Introduction

1.1 Image processing, analysis and understanding

Image processing is concerned with the manipulation and analysis of pictures by computer. In digital image processing a picture of a scene is usually represented as a two-dimensional, rectangular array of integers, or possibly a set of such arrays. Each element of the array, i.e. *pixel*, corresponds to one small area of the visual image, and the value stored there represents, usually, the light intensity measured for that area, quantised to one of a set of discrete *grey level* values. In applications requiring colour information several values are measured at each sample point, representing intensity measurements in a set of spectral bands; the image is then represented by a set of arrays.

The term 'image processing' can be used to encompass all sorts of activities involving pictorial data but is often used to refer only to the processing of pictorial information which produces results in the form of another image intended for human interpretation. Examples of such image processing applications include the removal of geometric distortion or blurring from an image, the improvement of contrast, etc.

A second major area of activity concerns the processing of pictorial data for autonomous machine perception. This is sometimes termed 'higher level' processing and may be described more specifically as image analysis, image pattern recognition, computer vision or image understanding. There is a large degree of overlap in meaning amongst these terms: Rosenfeld [1988], for example, considers them to be synonymous.

The view taken here, however, is that important distinctions can be made between image analysis or pattern recognition on one hand and image understanding or computer vision on the other. This stance accords with that of Brady [1982], who points out the following significant differences:

> "... • Pattern recognition systems are concerned typically with recognising the input as one of a (usually) small set of possibilities. Image understanding aims to construct *rich descriptions* that cannot be enumerated in advance but need to be constructed for each individual scene. Three-dimensional scenes, viewed from an arbitrary location, give rise to a wide variety of occlusion (overlap) relationships. One can hope to compute descriptions of three-dimensional layout but not to recognise it as an instance of a small number of stored prototypes...

• Most significantly, pattern recognition systems typically operate directly on the image. Image understanding approaches to stereo, texture, shape from shading, and indeed most visual processes, operate not on the image but on symbolic representations that have been computed by earlier processing such as edge detection. ..."

This thesis shows that the analysis of rock sections can be viewed as an image understanding problem, so bringing a novel perspective on the subject, and explores the possibilities of this approach.

1.2 Thesis overview

This thesis considers the application of image analysis in petrography - the study of microscope images of rock sections - and investigates the possibilities for advancing existing techniques by introducing feature extraction and analysis capabilities of a higher level than those currently employed. The approach developed takes petrographic image analysis beyond pattern recognition methods based on predefined measurements and towards image understanding, where the aim is the construction of meaningful descriptions of the rock section.

The principal contributions of this thesis relate to the fundamental problem of assigning the 'ownership' of crystal boundaries in igneous rock sections: the interface between two adjacent crystals is usually characteristic of only one crystal, which formed first, and 'imposed' upon the other, which crystallised later and grew against or around the existing crystal. The effects seen in cross-sections due to this crystallisation style are analogous to occlusion effects in two-dimensional images of three-dimensional scenes: just as parts of an object are obscured by other objects standing between it and the observer, and the region(s) representing the object are seen as bounded at least in part by the outlines (occluding boundaries) of the intervening objects, so it is that the regions representing a later forming crystal may be bounded at least in part by the characteristic boundaries of its earlier forming neighbour.

The importance of this problem is clear: meaningful descriptions of crystal form and inter-crystal relations cannot be derived until the ownership of boundaries has been established. The identification of characteristic boundaries can also indicate the crystallisation sequence of minerals in a rock and so provide petrological information.

Through an examination of the reasoning used by human observers the assignment of boundaries problem is shown to be a relatively complex image analysis task. Image features which reveal the ownership of boundaries are established. The design of a suitable petrographic image analysis scheme is then outlined and the implementation of key components considered in detail. In particular, novel contributions are made in the form of algorithms for shape and symmetry analysis. These analyses contribute both to the solution of the boundary assignment problem and to the construction of geologically useful descriptions of crystal form.

The analysis scheme presented employs grouping principles such as collinearity, parallelism, symmetry and continuity, so providing a link between this study and more general work in perceptual grouping and intermediate level computer vision. Consequently, the techniques developed in this study may be expected to find wider application beyond the petrographic domain.

1.2.1 Thesis structure

Chapter 2 provides a brief introduction to petrography and the role of microscopy. A review of image analysis in petrography is then conducted. This review establishes the importance of quantitative methods in petrography and examines how image analysis is used to automate such techniques. The extension of current capabilities is discussed and preliminary proposals for research are made.

Chapter 3 investigates the geometric features seen in crystalline rock sections, concerning the description of individual crystals and inter-crystal relations. Petrological background information is given to explain why certain features occur and are significant, and the terms appropriate for their description are discussed. The fundamental problem of 'boundary assignment', described above, is considered in detail. Key image features which reveal the ownership of crystal boundaries are established; a set of explicit analysis rules is presented. These rules indicate the image analysis tasks which must be tackled if the assignment of boundaries problem is eventually to be automated, motivating subsequent research. The design of a image analysis scheme based on these principles is outlined. The following chapters consider the implementation of important elements of this analysis scheme.

Chapter 4 describes the creation of a database of basic spatial information derived from a rock section image. Problems of data extraction and representation are considered. As petrographic image analysis proceeds this database will be both interrogated to supply the input data for the various analysis routines and updated through the addition of the new information which results.

Chapter 5 examines the problem of identifying crystal faces on a crystal section. Existing algorithms for deriving polygonal approximations are shown to be inadequate for this task. A new algorithm for multiscale contour analysis is then presented in detail. This algorithm can identify significant linear and near-linear structure on a contour over a wide range of scales, avoiding the use of arbitrary error thresholds, producing a hierarchical description termed the 'face tree'. Further, the algorithm is able to cope with variations in the optimal scale around the contour. This is a novel feature.

Chapter 6 considers the development of algorithms for the symmetry analysis of crystal profiles, taking as input the results of the multiresolution contour analysis of the previous chapter. The symmetry analyses implemented are concerned with 'internal' symmetries which may be deduced via the orientation properties of faces, but which need not produce a crystal which is symmetric in the normal sense of the term. Existing analysis techniques are therefore not applicable and new techniques are developed.

Chapter 7 presents an overview of the petrographic image analysis system of which previous chapters form a part. The state of implementation achieved for the component analysis modules is described. A possible control strategy is outlined, indicating how the the overall system may be realised.

Chapter 8 reviews the work of the previous chapters and the contributions made. Directions of further developments are discussed.

CHAPTER 2

Image Processing in Petrography

2.1 Introduction

This chapter considers the role of computer image processing techniques in petrography - the study of rock micrographs. In the terminology of the previous chapter, the emphasis is on higher-level processing, i.e. image analysis and understanding.

A brief introduction to petrology and petrography, including the role of microscopy, forms the first part of the chapter. This provides some essential background information for non-specialists. The major part of this chapter is a review of published work in petrographic image processing. This review considers the question of what sort of information is sought from rock micrograph images, and why, as well as examining how this information may be obtained. The importance of petrographic image analysis is demonstrated, and some obstacles to advancement are identified.

The possibility of extending the application of computer methods in the petrographic domain through the use of higher-level image analysis techniques is considered in the concluding section of this chapter, motivating further research.

2.2 Petrography

Petrology is the study of the origins, constitution and structure of rocks. Petrology therefore occupies a central place in geology, which is a broader, multifaceted discipline. Petrography, literally the description of rocks, is closely related to petrology since a meaningful description of a rock requires a knowledge of petrological principles. The term petrography is usually used to mean the study of rocks in thin section. This section provides more information on these topics.

2.2.1 Petrology

Rocks can be divided into three categories: igneous, sedimentary and metamorphic. The classification is on the basis of the conditions in which they formed. The American Geological Institute's Glossary of Geology [Bates & Jackson 1980] defines the groups of rocks as follows:

Igneous rock: a rock that solidified from molten or partly molten material, i.e. from a magma.

Sedimentary rock: a rock resulting from the consolidation or lithification of loose sediment or the chemical precipitation from solution at or near the earth's surface; or an organic rock consisting of the secretions or remains of plants and animals.

Metamorphic rock: any rock derived from pre-existing rocks by mineralogical, chemical, or structural changes, especially in the solid state, in response to marked changes in temperature, pressure, and chemical environment at depth in the earth's crust; i.e. below the zones of weathering and cementation.

This classification scheme is not clear-cut, for nature is a continuum, and borderline or transitional cases exist.

Chapter 3 contains further petrological information concerning crystalline rocks. The following brief comments can, however, be usefully made here.

The objective in petrology is an understanding of the forces and events which shape a rock's history. Equipped with this understanding one may then attempt to unravel the biography of any given specimen. For an igneous rock it may be possible to deduce, say, the composition and crystallisation history of the parental magma body. For a sedimentary rock the source of sedimentary material and the nature of the transportation and deposition process might be deduced. For a metamorphic rock the pre-metamorphism state of the rock and the nature of pressures and temperatures to which it has been subject might be inferred.

There is a second form of investigation into rock properties, which is concerned solely with the physical properties of the rock, *per se*. This may be termed an engineering interest. For example, the porosity and permeability of a petroleum reservoir sandstone may be of interest, or there may be a wish to study the nature of a fracture network in a crystalline rock, perhaps as part of a civil engineering investigation.

2.2.2 Rock micrographs

Microscopic methods play a crucial role in the study of rocks. The bulk of micrographs which are studied may be divided into three categories:-

1) Electron micrographs.

These are created by, for example, backscatter or scanning electron microscopy (BSE and SEM).

2) Transmitted light micrographs.

This is the traditional, the most convenient, and by far the most common, means of detailed visual examination of rock samples. If a slice of rock is ground down to a thin enough wafer, called a thin section, most of its constituent minerals become transparent. The thin section is then examined in polarised light, using a special mineralogical microscope.

3) Reflected light micrographs.

Here a small sample of the rock is polished smooth and studied in reflected light. This technique is usually used only for the study of opaque minerals (usually ore minerals). Perhaps surprisingly, the technique has much in common with transmitted light microscopy.

Of the three categories given, transmitted light micrographs are probably of the greatest importance in geology, particularly in the study of crystalline rocks. All geologists are familiar with such micrographs: they form the basic medium for geological instruction and research.

2.2.3 The mineralogical microscope

The mineralogical microscope is characterised by its use of polarised light. Light is a form of electromagnetic radiation, and may be thought of as consisting of electric and magnetic vectors which vibrate at right angles to the direction in which the radiation is moving. In this discussion of mineral optics only the electric vector need be considered, since it is the interaction of the electric vector with the electrons in minerals which affects the behaviour of light, and the forces arising from the magnetic vector are negligible in comparison.

Ordinary light, like that coming directly from a light bulb or the sun, vibrates in all directions normal to the propagation direction - it is unpolarised. When the vibration of the electric vector is constrained within a single plane the light is said to be plane-polarised. (The magnetic vector always vibrates at right angles to the electric vector.) There are various types of polarisation, but plane-polarised light is the most important for crystal optics.

In the mineralogist's microscope the substage light source has a polarising filter - the polariser - so that the mineral slice is viewed in plane-polarised light. The stage is not fixed, but may be rotated, and a graduated scale allows precise measurement of this rotation. Further, there is an additional polarising device - the analyser - in the microscope tube which may be slid in or out of position as required.

The analyser and polariser are arranged so that the vibrations they transmit are mutually perpendicular. These directions are also those of the crosshairs in the eyepiece. With the analyser in position observations are said, for historical reasons, to be made 'between crossed Nicols' or using crossed polars (XPL); with the analyser out of position observations are said to be in plane-polarised light (PPL).

2.2.4 Crystal optics

On the basis of its underlying symmetry, every natural crystal can be classified into one of seven symmetry groups: Cubic, Orthorhombic, Tetragonal, Trigonal, Hexagonal, Monoclinic or Triclinic. A full treatment of crystal symmetry may be found in texts such as those of Phillips [1971] or Cox *et al.* [1974], for example; some details are also given in the following chapter.

Cubic crystals are optically isotropic, as are many non-crystalline substances such as liquids and glasses. In isotropic materials the velocity of light is the same in all directions, and isotropic crystals will appear dark between crossed polars. This is because they do not affect the polarisation direction of the light from the polariser, and the light passing through the sample is then completely absorbed by the analyser - since polariser and analyser have perpendicular directions and the light reaching the analyser will have no component of vibration in its vibration direction.

Most crystals are not cubic, however, and so are anisotropic: their physical properties differ if measured in different directions. Further, when a ray of light enters an anisotropic crystal it is in general split into two rays - called ordinary and extraordinary - of different velocities and which are plane-polarized in mutually perpendicular planes. The positions of the vibration directions are fixed with respect to the crystal. Thus rotating a specimen on the microscope stage results in the rotation of these vibration directions.

When one of these vibration directions is parallel to the plane of polarisation of the light from the substage polariser, the light will pass through the crystal using this one vibration direction only, since it has no component of vibration in the direction normal to this. This light will be completely absorbed by the analyser, and the crystal will appear black. This phenomenon is known as extinction.

When the polariser's plane of polarisation is not parallel to one of the vibration directions of the crystal, the light is transmitted in two components, using both these directions. This phenomenon is known as double refraction.

If an anisotropic crystal is viewed in a non-extinction position with the analyser in place it will, in general, appear illuminated and coloured. These colours are called the polarisation colours, and are created as the two component rays recombine and interfere in the analyser, and is due to the phase difference of the rays. This arises from four factors, but only one factor is not constant for all grains in the thin section:

the difference in refractive index between the two rays. The difference between the maximum and minimum refractive indices of a mineral is known as its birefringence. With a knowledge of the section thickness (usually a constant 0.03mm) the birefringence value can be inferred from the observed colour by using a colour chart.

As the crystal is rotated the colours may therefore change, and the crystal will become dark in four positions ninety degrees apart as one of the vibration directions becomes parallel to the vibration direction of the polariser, as described above. Determining extinction positions permits the identification of vibration directions in a minerals grain. This then allows the study of the two modes of transmitted light individually.

In some minerals a change of colour occurs when the crystal is rotated in plane-polarised light (without the analyser). The effect is known as pleochroism and is due to the ordinary and extraordinary rays being absorbed differently on passing through the mineral. The effect can be quite marked for some minerals, with colour changes from green to brown or blue to green, for example.

2.2.5 Features of mineral sections in transmitted light

The appearance of minerals in thin section in PPL and XPL will now be considered, with some comments on the implications of these features for automatic image processing. These features are more easily seen than described; reference may be made to pictorial atlases such as those by MacKenzie [MacKenzie & Guilford 1980, MacKenzie *et al.* 1982] for illustrations.

Morphology

Crystals are usually bounded by flat surfaces called faces, although some crystals rarely show faces and develop in rounded masses. The way various faces are developed is termed the crystal habit; some crystals show very characteristic habits. Crystal shape is really a three-dimensional property, yet it can be studied by means of two-dimensional slices or sections. Crystal faces, of course, appear as straight edges in sections. Several crystal cross-sections may have to be studied in order to get an accurate idea of the (three-dimensional) crystal form. Elongated cross-sections may represent truly elongated crystals, or perhaps platy, tabular ones. Finding a tabular section resolves the ambiguity. Mineralogists learn to construct useful three dimensional mental models from two dimensional sections, relating the optical and morphological properties of the crystal. This is a complex task, with the handling of many 'clues', such as the cleavages of sections (defined below), to infer the relevant relationships between various sections.

Crystal shape is a very important indicator of the identity of the mineral. In addition, the symmetry and form of crystals, such as the development of good crystal faces, may allow the petrologist to make deductions about conditions of crystallisation, for example [3.2].

Relief

The visibility of a mineral in thin section depends on the difference between its refractive index and that of the mounting medium, which is usually Canada Balsam with a refractive index of 1.54. The quality of visibility is termed relief. In minerals of high relief, details of surface texture, cracks and cleavage, etc are very obvious. Minerals of low relief appear featureless and may indeed be barely visible. Relief is said to be positive or negative, depending on whether the mineral's refractive index is above or below that of the balsam. The sign of the relief may be determined by the Becke line test. The Becke line is a bright line usually seen at the edge of the mineral; as the microscope tube is moved up out of focus this line moves into the medium of higher refractive index. Since light passes through anisotropic minerals as two rays there can be a variation in relief with orientation. These factors make the proper estimation of relief a difficult image processing task for an automatic system.

Cleavage

If the relief is sufficiently high, some minerals may show a series of fine parallel cracks. This is the mineral's cleavage - an expression of the natural planes of weakness intersecting the plane of the section. Depending on how the thin section has been cut, and the structure of the mineral, of course, there may be several sets of cleavage cracks visible, of various characters, or none at all. The angular relationships between sets of cleavages and between cleavages and faces are important for mineral identification, and since the cleavage reveals something of a crystal's internal symmetry, it can play a role in the identification of rational faces [3.5.2.3].

Colour in PPL

Some crystals appear coloured in PPL because of the absorption of certain wavelengths from the white light supplied by the sub-stage source. Different sections through a crystal may produce different colours, but the colours are independent of the section thickness. As mentioned above, these colours may also change as the section rotates in PPL, in the phenomenon of pleochroism.

The colour is often variable or patchy in quality, which complicates automatic measurement.

Colour in XPL

The phenomena of polarisation colours have been explained above [2.2.4]. These colours show much greater variety than the natural crystal colours (in PPL). The quality is even more patchy and variable than for PPL, the reason being that for minerals with high birefringence small thickness variations can cause significant changes in the phase difference. In some cases the absorption colour may mask the polarization colours.

The extinction phenomena seen in XPL are important because they allow the vibration directions to be deduced. Relating the crystal optics to the morphology, for example by noting the angular relation between extinction direction and cleavage direction, is also used in mineral identification. This test can distinguish between some sections of biotite and hornblende, for example.

Twinning

When a single crystal consists of two or more parts in which the crystal lattice is differently oriented, it is said to be twinned. There are various types of twins. Simple twins consist of two segments which usually extinguish at different positions when rotated. Polysynthetic twinning, commonly seen in plagioclase feldspar, for example, results in numerous twin segments joined on parallel twin planes. The twin boundary is often invisible in PPL. (It may be observed that the junction between the two segments of a twin as seen in a thin section is usually a straight line, and is unlike a normal intercrystal boundary.)

2.2.6 Reflected light microscopy

The microscope system used for viewing sections in reflected light is very similar to that used for transmitted light. The incident light is plane polarised and is reflected into perpendicular incidence on to the specimen using a glass plate reflector or half-field prism. The microscope tube again holds an analyser which may be slid into position if required.

The phenomena seen in reflected light microscopy are analogous to those found in transmitted light microscopy. However, in contrast with studies of thin sections in transmitted light, the properties exhibited by each phase are qualitative and may not be sufficient for unequivocal identification. Minerals may show a variety of colours and apparent brightnesses. The brightness is more exactly termed 'reflectance' and is the ratio of the intensities of reflected light to incident light. This property can only be accurately measured with some additions to the standard reflecting microscope.

As with transmitted light microscopy, the optical properties of minerals may vary with orientation. The change of reflectance is the property of bireflectance, and the change of colour is termed reflection pleochroism. Anisotropic minerals may also be studied with the analyser in place. (Isotropic minerals remain in extinction in all orientations when the analyser is used.) Sections displaying anisotropy will be in extinction, or at least showing minimum brightness, in four positions ninety degrees apart. The colours exhibited with the analyser in place are termed anisotropic colours, and may be used as an aid to identification.

The morphology of crystals is described in the same terminology used for thin sections. Cleavage is less commonly observed in polished sections than in translucent thin sections. If three or more cleavages are present in a mineral parallel rows of triangular pits may be observed: this phenomenon is characteristic of certain minerals, notably galena. Twinning is also less obvious in polished sections. It may sometimes be evident from a change in the orientation of cleavages as they cross over the twin plane.

2.3 Image processing in petrography: a review

This review of published work on the use of computer image processing and analysis in the study of petrographs was undertaken at the start of investigations in the field and was brought up to date at the conclusion of the project. It is not exhaustive, but is considered adequately representative of the techniques which have been used and which are currently in use. Further relevant review material may be found, for example, in publications by Amstutz and Geiger [1972], Jones [1977, 1987] and Fabbri [1984, 1986].

It will be seen that computer image processing and analysis techniques are valuable in both petrological and engineering applications. In many cases the studies would be impracticable or impossible without automation. The limitations of image processing will also be considered. The possibilty of overcoming some of these problems will be considered in the following section [2.4], where proposals for research will be made.

In general, rocks are aggregates consisting of one or more phases. (A phase is a set of components having similar physical or chemical characteristics; this definition implies that voids or pores in the rocks may be treated as a phase.) The characteristics of the phases and their distribution in space reflect upon the processes which have formed the rock, and will also control its 'engineering' properties, such as the permeability of the rock or how well certain mineral components may be separated from the aggregate by milling. Thus there is a need to study the distribution of grains within the rock, and to characterise their geometrical attributes such as size, orientation and shape.

Since it is impossible to observe directly the three-dimensional properties of the individual components in an opaque rock, two-dimensional sections must be used. If the interrelations between the components are not of interest it may be sufficient to analyse the shapes of the individual grains. (Section 2.3.1 below considers the quantitative analysis of grain shape.) Very often, however, the spatial distribution of phases is of interest, in which case simple shape analysis will not be adequate.

Section 2.3.2 considers the basic ideas of mathematical morphology, which has been found to be an important tool in image analysis. The techniques of mathematical morphology were originally developed for petrographic image analysis, for the study of the spatial distribution of one particular phase within a rock, usually the pore network in a reservoir sandstone. One such analysis scheme will be described. The methods of mathematical morphology also support the analysis of inter-phase relations in multi-phase aggregates. Such techniques are considered in section 2.3.3.

Section 2.3.3 discusses the quantitative analysis of rock textures and of the spatial relations of phases in a rock. Examples are given of the use of such information in petrological studies and in the monitoring of mineral ore dressing. Some petrographical image processing and analysis systems will also be briefly described.

Section 2.3.4 considers what is perhaps the key bottleneck in automatic petrographic image analysis: the identification and labelling of the components in the image.

2.3.1 Shape analysis

This review section is concerned primarily with quantitative shape analysis methods devised for sedimentary particles, either viewed in a rock section or as unconsolidated grains. The description of mineral grains within the crystalline igneous and metamorphic rocks is considered in detail below [Ch.3]. (A mineral grain is a mono-mineralic fragment not necessarily freed from its matrix; a mineral particle is a free fragment that may or may not be mono-mineralic.) It may be noted that the descriptors developed for sedimentary particles are virtually the only shape parameters which have been generated by computer image analysis for petrographic studies. These can, of course, be applied to non-sedimentary grains if appropriate. For example, the Fourier descriptors developed by Ehrlich and Weinberg [1970] have been used by Ehrlich *et al.* [1972] in the petrogenetic analysis of a granitic gneiss.

In their study selected quantitative textural variables such as grain shape, and surface area of phase contacts per unit volume (estimated from the section simply by averaging the number of grain intersections of a random line of unit length [Underwood 1970]) provided information regarding reaction pathways and kinetics. Histograms were produced for rock samples taken along a traverse of the exposure, and examined visually. The textural variables demonstrated the presence of gradients and discontinuities in gradients, whereas compositional data taken from the same samples were largely unresponsive. One notable conclusion of this study was that textural variation contains information of a similar magnitude as compositional variation.

Grain shape has long been recognized as a fundamental property which may be used to characterise and analyse sedimentary rocks. Careful examination of the grains: their compositions, sizes and size range, shapes and shape variability, and of the rock structure can allow an unravelling of the rock's history. It may be possible, for example, to deduce the nature of the parent rocks (which provided the sedimentary material), the nature of the sediment transport and the depositional environment.

Geologists using only visual inspection of particles have been restricted to a qualitative description of the size and shape. Thus description of shape has often been limited to estimates of the angularity and sphericity. Estimates of angularity are given through the use of terms such as well rounded, subangular, extremely angular, etc., and similarly for sphericity. Understandably, there has long been a desire to quantify shape descriptions, so providing objective, numerical indices for analysis. Computer image analysis can provide detailed parameters to describe the shape of a single particle.

Early automated techniques concentrated more on the overall shape and measured simple proportions such as aspect ratio, perimeter to diameter ratio or the ratio of inscribed to circumscribed circles in the silhouette. Clark [1981] provides a review of more sophisticated endeavours.

The shape descriptors described by Clark are for two-dimensional grain profiles. The descriptors are considered in two groups: those which treat the grain as an outline, and those which treat it as a planar surface. The outline may be 'unrolled' and analysed through fitting a Fourier series. One such technique is that of Ehrlich and Weinberg [1970].

Clark also describes distributional approaches, where the distribution of chords, radii or angles are analysed, without regard to the sequential nature of the outline. When the grain profile is considered as a planar surface, the techniques applied include the derivation of 'equivalent ellipses', which are equivalent to the profile in that they share some dynamic characteristics, such as moments of inertia. There are also functional approaches which treat the profile as a binary-valued function on the plane which is then characterised by means of a function such as the two-dimensional Walsh function. Details need not be included here.

A more recent approach to shape analysis makes use of the fractal dimension of an outline. This approach is due to Mandelbrot [1982], and is now used in several disciplines. The length (P) of a curve, or grain perimeter, may be measured in terms of a number of steps (n) of fixed length (x): P=nx. As the step size x decreases, the apparent length P increases. Mandelbrot has shown that n may be expressed as a function of x: $n=\lambda.x^{(-D)}$, where D is interpreted as a dimension. This 'fractal' dimension of an outline is a measure of its ruggedness or crenellation, and provides a powerful alternative to Fourier methods for irregular particles, since it provides a single value estimate of form.

Orford and Whalley [1983] were perhaps the first to use the fractal dimension to quantify the morphology of irregularly shaped particles in sedimentological studies; Clark [1986] gives three techniques for implementing digital fractal analysis of particle shape; Dearnley [1985] illustrates the application of fractal dimensions in a study of the effects of resolution on the measurement of grain 'size' in automatic image analysis; Latham and Poole [1987] compare Fourier and fractal coefficients in the study of rock aggregate particle shape.

2.3.2 Mathematical morphology

Mathematical morphology is an approach to image processing based on set theoretic concepts of shape. Morphological operations have long been used in image processing. Matheron [1967] and Moore [1968] gave early examples of the use of mathematical morphology in petrographic image analysis; Watson [1975] presented an introduction to the theory, specifically aimed at workers in petrography. At first morphological operations were only developed for binary images, but the ideas were later extended to greyscale images. The development of mathematical morphology is largely due to J. Serra and G. Matheron, at the Centre de Morphologie Mathematique of the Ecole Nationale Superieure des Mines de Paris.

Serra's [1982] book formally presents the basis and theory of mathematical morphology; Haralick *et al.* [1987] have produced a more accessible tutorial paper. Their paper develops the basic relationships in binary morphology and the extensions of these relationships in greyscale morphology. The work need not be reproduced here. However, the important ideas of 'erosion' and 'dilation' are often referred to below, and so an explanation of these key concepts will be given.

Following this brief tutorial, an example petrographic image analysis system which uses mathematical morphological methods will be considered. The application is the analysis of pore networks. This is the morphological analysis of a single phase, with no requirement for the study of inter-phase relations. Mathematical morphology techniques may also be used for the analysis of multi-phase aggregates, as will be considered in the following section [2.3.3].

2.3.2.1 Erosion and dilation

'Dilation' (also termed 'dilatation') is the morphological transformation combining two sets of points (or pixels) in the plane, using vector addition of set elements. If A and B are the sets of points, then the dilation of A by B, denoted A⊕B, is the set of all possible vector sums of pairs of elements, one from A and one from B:

$$A \oplus B = \{ c = a + b \mid a \in A \ \& \ b \in B \}$$

The dilation operation is commutative but in practice the two sets of points are handled differently. A is considered as the image under analysis and B, referred to as the structuring element, can be thought of as a shape parameter in the transformation. Commonly the structuring element is a simple set of points such as a 3X3 square of pixels. Dilating an image with such a structuring element adds an outer layer of pixels to the image and in general simplifies or smoothes the shape.

'Erosion' is the morphological dual to dilation. The erosion of A by B, denoted AΘB, is defined by:

$$A \Theta B = \{ c \mid c + b \in A \text{ , for every } b \in B \}$$

Equivalently, $A \Theta B = \{ c \mid \text{for every } b \in B \text{, there exists an } a \in A \text{ such that } c = a - b \}$

The duality arises because eroding A by B is exactly equivalent to dilating the complement of A by the transpose of B, then taking the complement of the result. The transpose of B is the set formed by taking the negative value of every element in B. If B is left unaltered by a rotation of 180 degrees about the origin, as are many commonly used structuring elements, then it is equal to its transpose .

Erosion and dilation are usually applied in pairs. Erosion followed by dilation is called 'opening'. Dilation followed by erosion is called 'closing'. The opening of B by structuring element K is denoted by BOK: $BOK = (B \Theta K) \oplus K$.

The closure of B by the structuring element K is denoted by B•K: $B \bullet K = (B \oplus K) \Theta K$. In general, opening or closing an image alters the image. For example, opening an image with a disk structuring element smoothes the outline, breaks narrow isthmuses and eliminates small islands and sharp corners in the profile. Closing an image with the same structuring element smoothes the outline, fuses narrow breaks and notches in the outline and eliminates small holes. Thus closing is often used to 'clean' an image.

2.3.2.2 Pore complex analysis

The theory of mathematical morphology stems from the problem of characterising porous media, particularly oil reservoir rocks. Matheron's fundamental work on the theory of porous media [Matheron 1967] contained, for example, an analysis of the notions of granulometry and permeability, and demonstrated that the concepts of erosion and dilation, and of opening and closing sets by structural elements, could be used to describe in probabilistic terms the morphology of pores and grains in thin sections.

This work on pore analysis has been continued by several workers. Delfiner [1972], for example, analysed the porosity of sandstones using measurements taken from a thin section micrograph. One commercially successful system for pore analysis, which uses relatively simple mathematical morphology techniques to derive estimates of the (three-dimensional) reservoir properties of a sandstone from a two-dimensional section, has been developed at the University of South Carolina by Ehrlich, Crabtree and co-workers [Ehrlich *et al.* 1984, Crabtree *et al.* 1983]. This system is considered as an illustration of the technique.

The rock is viewed simply as a two phase aggregate: pores and non-pores. The binary pore image is characterised through the use of repeated openings, using increasingly large structuring elements. The resultant images are analysed so as to generate 'spectra' (see below). These are in turn analysed by pattern recognition and classification algorithms.

The first erosion-dilation cycle uses a 3X3 pixel structuring element which strips off one layer of boundary pixels and then, if a 'seed' pixel remains, adds one layer of pixels. This single cycle begins the removal of any roughness on the object and destroys any portions of the pore image of two or less pixels in width. The next cycle performs two successive erosions of the original image followed by two successive dilations, if a seed remained. Successive cycles of increasing magnitude continue until the last erosion destroys the seed pixel(s), completely erasing the object.

Ehrlich and Crabtree exploit the fact that a given amount of erosion followed by the same amount of dilation need not restore the image to its original state. Small pores may be lost through the loss of their seed pixels and, similarly, surface roughness elements lost in the erosion will not be restored. By comparing the original image with its successive openings, frequency distributions, termed pore-complex spectra, can be constructed showing the proportions of total image porosity lost at each cycle. The algorithm checks to see whether the loss of a pixel is due to the loss of an entire pore or to the loss of a portion of a pore. Thus three spectra are produced: total pore loss per cycle, pores lost per cycle (a measure of smooth pore size) and pore roughness lost per cycle. Since each cycle operates upon an integral number of pixel layers, each cycle is related to an absolute spatial scale defined by the microscope magnification and the size of the pixel grid. The spectra can thus be given a linear scale in units such as microns. The result is a partition of any given pore into two parts: roughness and size-smoothness.

The pore spectra generated are used in the subsequent classification phase. Here the pore complexes are separated into sub-complexes based upon spectral signatures, in an 'unmixing' process. The authors view this work as one component of a future 'robot petrographer'.

2.3.3 Quantitative image analysis

The need for petrological studies to make use of quantitative textural information has been expressed for many years (see, for example, the papers by Kretz [1966], Amstutz and Giger [1972] and Ehrlich *et al.* [1972]). However, since such information cannot be obtained by simple visual inspection, the emphasis in petrology has traditionally been placed upon mineralogical and compositional variables, which can be more readily estimated by eye or by comparatively simple laboratory analyses. Due to the development of stereology and, more recently, of computer image processing capabilities, the situation has changed.

Developments in image analysis have also made it possible to measure accurately mineral properties which predict and define the behaviour of ores during mineral dressing [Petruk 1982]. Image analysis can therefore be used in monitoring the state of mineral processing and in the design of appropriate processing regimes.

Some example applications of image analysis in petrology and process mineralogy will now be described. Following this, some image analysis systems will be considered.

2.3.3.1 Applications

Studies of the geometric properties of a three-dimensional complex from measurements made on two-dimensional sections or projections are, by definition, stereological. Stereological methods are indispensible tools in metallurgical and ceramics studies; they have become increasingly common in the study of rocks. The study of mineral aggregates, such as an igneous rock, is slightly more complex than metallurgy, since almost always more than two components are of interest.

Stereological principles have been established relating measurements on planar sections of a complex to its three-dimensional properties; a comprehensive reference text is that of Underwood [1970]. The derivation of many of the stereological relationships between two-dimensional images and three-dimensional specimens that they represent may be complicated, but the application of these relationships often involves only simple arithmetical procedures.

One such stereological result relates the relative volumetric proportion of mineral phases of rocks to planar section measurements. It turns out that for randomly distributed phases the volume proportion of the phase is equal to the areal proportion of the phase in a random cross section. Further, the areal proportion is equal to the linear proportion along a random line on the section, which is equal to the point proportion. Thus volumetric proportions of minerals can be estimated by counting the proportion of 'hits' on the minerals in a random sample of points on a section [Underwood 1970, Jones 1977].

Kretz used stereological principles in his study of a pyroxene-scapolite-sphene granulite (a high-grade metamorphic rock) [Kretz 1969]. This was one of the first investigations to use the statistical analysis of grain profiles for quantitatively relating their geometrical attributes to nucleation and crystallisation processes which could be modelled for the rock. The study was particularly concerned with establishing suitable definitions and tests for homogeneity and randomness which will allow one to decide objectively whether or not the small clusters of crystals found in some rocks, for example, may be attributed to chance. Kretz produced a suite of tests for homogeneity, the randomness of distribution of minor component minerals and of major component minerals, and of crystal associations involving minor component minerals and involving host and inclusions.

Concerning the petrological side of his investigation, Kretz concluded that the rock was homogeneous and its crystals randomly distributed. Asking what the implications were for the nucleation and growth of the crystals in the rock, Kretz modelled two phases nucleated instantaneously and at random, with the 'crystals' expanded at a uniform rate in all directions until impingement was complete. Applying some of the tests used on the real data gave results indicating similar levels of randomness. By analogy it was suggested that in the test section the sphene, scapolite and pyroxene crystals nucleated at random and the nucleation site for any crystal was not determined or influenced by any other crystal in the neighbourhood.

Flinn [1969] used techniques closely similar to Kretz's in a study of grain contacts in gneiss (a metamorphic rock). This study revealed a non-random distribution of grains in the rock: contacts between like phases occurred less frequently than would be expected if the grain types had been randomly distributed. Flinn concluded that during strong recrystallization a state approaching minimum interfacial energy is attained by reducing the relatively high energy contacts between unlike phases. The question of how this is actually achieved is then raised. This may take place by the diffusion, nucleation and growth of unlike phases in boundaries between like phases or, more likely, a physical migration of the grain boundaries.

Similar studies had previously been undertaken in attempts to distinguish igneous from metamorphic granites, notably by Vistelius. Vistelius and colleagues established the use of statistical models in petrology, as part of the greater thesis that mathematical methods are necessary in geology for constructing models of geologic processes i.e. for testing agreement between geological constructions or hypotheses and observational data [Vistelius 1967, Dvali *et al.* 1970].

One of the earliest of such statistical techniques to be developed was the use of stochastic modelling. For example, a stochastic model of phase relationships can be built up through observing the sequences of phase transitions along linear traverses of rock cross-sections. Whitten and Dacey [1975] used the methods of Vistelius to analyse the same granulite studied by Kretz [1969]. They found that it possessed first and second-order Markov properties. That is, the composition of a grain was controlled by the composition of its neighbours. (A discrete parameter stochastic process has the N-order Markov property if its next state depends only on the current state and the previous N-1 states: an explanation of these terms may be found, for example, in the text of Hoel *et al.* [1972].) This property reflects petrogenetic factors which are still under investigation.

Just as measurements of the size and shape of the grains in a metamorphic or igneous rock may yield significant petrological information, similar measurements can be useful in process mineralogy - the study of the behaviour of mineral ores during dressing. Measurements of the size of individual grains and/or the size distributions are important because sizes determine the degree of liberation of minerals during beneficiation (the processing of ores in order to improve their properties). Knowledge of the size distribution therefore allows for prediction of liberation characteristics and the use of an appropriate, optimised processing regime.

Pong *et al.* [1983] described the use of the GIPSY image processing system [Haralick *et al.* 1981] in process mineralogy: performing shape analysis, deriving size distributions and in liberation studies. The 'degree of liberation' is a stereological measure of the extent to which mineral particles occur as individual single phase units in a crushed sample, determination of which is vital during mineral processing. To monitor processing, samples are taken, sized, cast in an epoxy matrix, ground and polished, and then examined by reflected light microscopy.

The degree of liberation of a mineral component is defined by a simple formula involving the total area of the component occurring as free particles, the total area of component occurring as locked particles, and the locking factor of the mineral. A region is considered locked if it is adjacent to any other non-background region of different type, and is considered to be free otherwise.

Liberation computations were facilitated by the construction of a region adjacency graph (RAG). This is a graph in which the nodes represent the regions in an image and adjacencies between pairs of regions are recorded by arcs between the corresponding graph nodes. Once the RAG was obtained, locked and free particles could easily be determined by comparing the mineral types of pairs of adjacent regions. In addition, the RAG and property lists also allowed computation of the percentage of a phase which occurred as locked particles, the percentage of particles which were locked, and the correlation of each mineral species with each other mineral species.

2.3.3.2 Example image analysis systems

The development of statistical, stereological methods led to a great saving in human effort: point-counting is easier than measuring areas. Using an image in appropriate digital form these operations could be automated relatively easily. Further, many stereological parameters could be measured directly and accurately, removing the need to rely on statistical estimates. It is the case that the development of specialised image analysers has led to a return to the direct measurement of many stereological parameters such as area, length of contacts, or shape parameters [Fabbri 1986]. In addition, modern image analysis techniques, for example the shape descriptors discussed above [2.3.1], can provide information which standard stereological techniques cannot.

There are many image processing systems now available. Some have been specially tailored for petrographic image analysis; others are general purpose systems. The General Image Processing System GIPSY, a minicomputer system developed at Virginia State University [Haralick *et al.* 1981], as its name states, supports a range of image processing and analysis algorithms with general applicability. This system has been used successfully in process mineralogy [Pong *et al.* 1983] and in petrological studies [Craig *et al.* 1982].

Among GIPSY's capabilities are filtering, classification, geometric spatial clustering, morphological operations, region growing and property file generation. Measurements include area and perimeter, grey level information (maximum, minimum, mean and variance), the centre of mass, elongation and angles as measures of the shape and orientation, and a circularity shape measure. Elongation is obtained as the ratio of the lengths of the major and minor axes of the best fitting ellipse of the region. The circularity (or ellipticity) is given by the ratio of the mean to the standard deviation of the distances from the centre of the figure to its boundary points [Haralick 1974]. How GIPSY has been used to automatically identify and label the crystals in a polished section image is described below [2.3.4].

Some systems for petrographic image analysis perform stereological analyses by simply automating the traditional methods such as point counting, albeit with additional image processing capabilities. Examples are AMBA/R [Oettel & Ohser 1986] and the IBAS system [Köditz 1981]. Other systems produce stereological measures via the methods of mathematical morphology. Morphological functions were implemented in hardware and software in the Leitz TAS image analyser. Nawrath and Serra [1979a, 1979b] describe TAS and its use in the analysis of free and locked grains for mineral liberation studies.

A well-documented example of an image analysis system which uses morphological operations to obtain stereological measurements is the Geological Image Analysis Program Package GIAPP [Fabbri 1984]. This uses what has been called a picture processing approach to stereological problems [Fabbri & Kasvand 1980].

GIAPP

The GIAPP system analyses multiphase rock micrographs in terms of geometrical probabilities associated with the binary patterns obtained from the different phases. GIAPP is based on binary image processing; multiphase images are handled as a set of binary images in registration. Although used primarily for the analysis of rock micrographs, GIAPP has also been used in the analysis of map data.

The typical thin section image presented to GIAPP consists of boundary lines separating different areas, representing the crystals in the section. These may be given phase labels by the operator. The basic processing capabilities in GIAPP are:

1) pixel-by-pixel logical operations between binary images;

2) the mathematical morphological operations of erosion and dilation;

3) spatial shifts and comparison, typified by auto- and cross-correlation;

4) procedures for creating graphs (tables) of spatial interrelationships of areas;

5) procedures for preparing the data for analysis, e.g. automatic area labelling, interactive editing routines.

These facilities allow the analysis of relations between different phases. For example, in order to find the contacts between two phases the corresponding binary images can be dilated and then logically ANDed together. The resultant image region(s) represents points on the border of both phases, i.e. the interface regions. These regions may then be analysed further.

An example of the use of such techniques was given by Fabbri [1984, Ch.10] when he reproduced and extended Kretz's [1969] petrological studies, using GIAPP. Fabbri used exactly the same rock section image which Kretz had used, but in a digitised form.

In Fabbri's study applications were demonstrated involving the computation of the area and perimeter of the grain profiles, measurement of grain profile contacts and of their distribution, determination of the orientation of grain and grain-cluster profiles, and computation of the geometrical covariance function of the fabric. The applications explained how to study a rock fabric systematically and provided new geological tools for detecting the presence of, and describing the type of, crystal shape anisotropy or crystal cluster anisotropy which create foliations in the texture of crystalline rocks. The study thus provided a concrete example of the power and relative convenience of automatic image analysis.

Fabbri *et al.* [1983] demonstrated the use of GIAPP's routines for computing tables of adjacency relationships in a petrological investigation. From the sets of tables of adjacency information generated by GIAPP the estimated probabilities of the grains to be 1-step to N-step adjacent to any other grain could be computed. Particular two-dimensional sequences in a granulitic rock were extracted. These revealed hidden distributions and agglomeration structures of the grains in which reformation and crystallisation history may not have obliterated an original structure precedent to the last crystallisation events. Thus GIAPP may be used in investigations of phase relationships, such as those discussed above [Vistelius 1967, Whitten & Dacy 1975]. Similarly, GIAPP's facilities for analysing adjacency relationships could be used in process mineralogy, for example in liberation analyses [Petruk 1982].

2.3.4 Approaches to image segmentation

A major bottleneck in the application of automated computer image analysis is achieving a suitable segmentation of the data, i.e. finding the boundaries of the crystals or grains and, possibly, identifying the actual phases present. The GIAPP applications described above, for example, all used digitised images of grain profiles which were prepared from tracings or by a similar process, and all phase labelling was done interactively, i.e. under user control. Fabbri [1984, p114] commented that this remained the optimal scheme and "...it is not yet possible for automatic scanning devices to capture and process sufficient information for satisfactory phase recognition and extraction".

This need for a high level of user intervention slows down analyses and acts against the wider use of computer methods, so that they remain reserved for the more complex problems only. Applications of image processing in petrography which confront the segmentation problem have been found, however. The nature of the segmentation task may vary with the application. Segmentation problems of various levels of difficulty have been identified; in order of increasing complexity these are:

1) separate the pores/holes from the solid phases;

2) isolate the separate grains in the solid phase, with only one phase;

3) isolate the separate grains in the solid phase, with several phases present;

4) identify/label the separate grains.

This is only a generalisation of the ascending difficulty involved: the nature of the phases present is probably the major control. The examples found fall naturally into the complexity categories given above.

'Level zero'

It is possible to avoid the segmentation problem by treating the input image simply as a grey tone texture and making no attempt to identify or isolate grains and pores in the image. One early example of this was the investigation by Haralick and colleagues [Haralick & Shanmugan 1973, Haralick *et al.* 1973] into the automatic classification of sandstones by texture. The techniques used could be applied equally well to the classification of other pictorial data, such as aerial photographs and satellite images [Haralick *et al.* 1973].

Up to 28 parameters (such as mean and range entropy, contrast, correlation.) were derived from grey tone co-occurrence matrices. The co-occurrence P(i,j) of grey tones i and j for an image is defined as the number of pairs of pixels having grey tones i and j, respectively, and which are in a fixed spatial relationship, such as a fixed distance or a fixed distance and angle. The parameters derived were not statistically independent and subsets could be selected, so providing feature vectors which would, it was hoped, characterise the rock types. Classification proceeded using piecewise linear discriminant function methods. One study used 5 categories of sandstone; 143 rock samples as training data and 100 as test data. 89% accuracy of classification was reported.

Some of the statistical measurements are hard to appreciate intuitively. Further, according to Lin and Cohen [1982], a study of the autocorrelation function of texture density indicates that the few parameters reflecting physical properties which can be extracted are more easily measured directly, and the autocorrelograms and power spectra of different images are in general less different than the images themselves.

'Level one' segmentation

The pore analysis scheme developed by Ehrlich, Crabtree *et al.* [2.3.2.2] presented the pre-processing problem of accurately segmenting (dye-impregnated) sandstone micrographs, where only a binary segmentation was required. This task was not straightforward in practice, as has been reported by Crabtree *et al.* [1984].

The rock specimens were impregnated with blue epoxy before preparation of the thin sections. Since the rocks never contained naturally occurring blue minerals, this process assisted segmentation. Red, green and blue filtering gels were used to derive intensity, saturation and hue values for each pixel. By checking each pixel's values against predetermined limits for these parameters, the image could be segmentented into pore and non-pore. This created a simple binary image for use in the pore analysis.

Crabtree reported that most conventional segmentation techniques using thresholding on histograms failed because of problems associated with the high light intensities required for the microscopy and because of gradational boundaries caused by shelving effects. Successful segmentation was finally achieved by modelling digital filters on the human perception of the colour of pore pixels.

'Level two' segmentation

The work of Phillips *et al.* [1983] involved the segmentation of a sandstone micrograph into pores and non-pores, with the further step of finding the boundaries of the grains in the non-pore regions. The micrographs in this case were created by electron microscopy. However, low magnifications were used giving micrographs similar to optical images, and the processing techniques employed are of interest. The long-term aim of the project was the determination of the three-dimensional geometry and topology of the grains and pores in mineral specimens. The three-dimensional shapes of the pores were to be constructed from segmented two-dimensional images using a three-dimensional connected components analysis process. The initial phase of the investigation considered here was concerned only with determining the two-dimensional shapes of the pores and grains in a given cross-section of the specimen.

The initial segmentation into pores and non-pores was achieved by thresholding the image, after smoothing. The threshold value was selected by picking the deepest concavity point on the grey level histogram. Small regions were then eliminated, presumably by merging them into the neighbouring region with which they shared the longest border. The image obtained from this initial segmentation was then used as a mask to clear the background in preparation for the grain extraction step. The cracks in the image might have widely different contrasts, thicknesses and shapes, confounding the use of simple line detectors. Again the grey level histogram was clearly bimodal, facilitating thresholding. The threshold used was based on the same concavity point, with a slight bias introduced. Next the cracks found were thinned to one pixel thick, and end-of-crack pixels were marked. A crack extension procedure was then used to recover some lost crack pixels. Grain extraction required linking of the edges and cracks surrounding each grain. A multiresolution technique based on the idea of processing pyramids [Levine 1980] could be used to link the cracks to produce smooth, connected grain borders.

'Level three' segmentation

The work of Montoto and colleagues [Montoto *et al.* 1978; Montoto 1978, 1982] in the analysis of micrographs of crystalline rocks is noteworthy, since these were the first studies to attempt to segment such images in a 'natural' manner, i.e. by extracting the grains/crystals as perceived by a geologist. Both SEM and polarising optical microscope images were dealt with. It may be noted that the authors reported a failure to find any related studies in their literature search undertaken in 1978.

Digital image processing was employed on transmitted light micrographs with the aim of quantifying textural and fractographical parameters (relating to the system of cracks and fractures). The broader aim was the extraction of petrophysical properties such as mechanical strength, compressibility, resistivity, etc., which are directly related to petrographic characteristics. The use of multiband images, created through the use of colour filters and different analyser and polarizer positions, was proposed in order to detect the boundaries between adjacent grains which exhibit similar properties under certain conditions. This should also have allowed discrimination between external grain boundaries and the fractures and cleavage cracks within grains.

The microscope images were recorded on black and white photographic negatives, which were then digitized to 256 grey levels. The first processing step was the extraction of the grains by a multiple thresholding or 'density slicing' technique, i.e. a set of threshold values T_0, T_1,..T_N were selected so that each grey level value in the image lay in some interval [T_i,T_{i+1}], and was reset to the lower threshold value. This produces a small set of discrete greyscale values in the image. The thresholds were selected in a visual examination of the image, following smoothing, so as to partition the image into its constituent grains. Four threshold intervals were found to be adequate for the test examples examined.

The mean smoothing process introduced many small regions near grain boundaries. These small regions were eliminated in a shrink and expand process (closure), which also smoothed the grain outlines.

A simple template-type edge detector was then applied which returned edge magnitude, and approximate direction for edge points of non-zero magnitude. This process incorporated line splitting so that in the output every grain had its own boundary, which helped the subsequent tracing algorithm. The edge directions were used to guide the line following algorithm which produced as its output circuits of pixels which, by definition, were the grain boundaries. These boundaries were coded in tables of pixel coordinates. These lists were then reduced in size by using a polygonal approximation. From these lists primary image parameters such as perimeter, area and centroid coordinates, and derived parameters such as compactness and other shape factors could be calculated. Additional processing steps for the study of microcracks need not be considered here.

As would be expected in this segmentation scheme, (simple and lamellar) twinned grains could be wrongly interpreted as collateral assemblages of grains with parallel boundaries. The authors solved this problem through use of images with different polariser positions, although no more details were reported.

The implication is that they could find images in which the twin components were not distinguished. The sample in question consisted predominantly of twinned plagioclase feldspar crystals; these twins have the property that in certain orientations in XPL the twin components become uniformly grey. These special orientations must have been sought out, which would have required special user intervention for each twin present. The PPL image, which shows the feldspars as uniformly transparent, would not be suitable since no discrimination would be possible between twin components and unrelated, but adjacent, grains.

The analogous problem where adjacent grains exhibit closely similar properties is more easily solved with the use of multi-images. The need for special intervention to find positions which differentiate the grains is reduced since the probability that two random orientations will both give two unrelated, contiguous grains a closely similar appearance is very small.

Montoto *et al.* [1978] and Montoto [1978] reported work on a method to detect twinning without the use of a multiband image. A summary of the method was later published [Bel-Lan & Montoto 1982] describing the use of specialised reasoning techniques. As before, a description module described each region of homogeneous intensity by a polygonal approximation of its boundary. A decision-maker algorithm used these descriptions to identify those regions which were twins, identified which twins belonged to the same mineral grain, and so constructed the final boundary description for every mineral grain. The process was accomplished by a dialogue between the classifier algorithm and the description module, which also provided information about length and slope of polygon sides, distance to other sides of candidate twins and the grey levels which characterised the original regions.

It may also be noted that Montoto *et al.* [1978] presented the following future possibilities for image processing in petrography: a) routine petrographic studies, under reflection and/or transmission polarizing microscopy; b) studies under SEM observing the finest petrographic details such as microporosity and microfractography; and c) point, linear or areal chemical analysis, by means of an X-ray spectrometer attached to the SEM.

The density-slicing technique used by Montoto *et al.* requires much user control, and also depends on the grain sections being relatively uniformly toned, and similar grains non-contiguous. A simplified version of the technique for phase identification was later reported [Allard & Sotin 1988], where the assumptions made about the image were even more demanding and unrealistic.

'Level four' segmentation : GIPSY

Pong *et al.* [1983] have described how the General Image Processing System GIPSY [2.3.3.2] has achieved the automatic segmentation of an image of an ore mineral assemblage, with subsequent grain identification. The images were produced by reflected light microscopy, since opaque minerals were being studied. The authors report that, previous to their work, most commercial systems for automatic grain segmentation made a threshold decision on reflectances on a pixel by pixel basis. Such an approach was highly susceptible to the noise inherent in the grains' reflectances. In contrast, the approach used with GIPSY was to first segment the image through the use of an edge operator to delineate grain boundaries. The grains were then identified using the *average* reflectance of all pixels in the region bounded by the edges, a value clearly less susceptible to noise.

Edge detection was accomplished using the facet model developed by Haralick and Watson [1981]. Under this scheme the intensity surface is assumed to approximate an ideal underlying model consisting of a patchwork of connected regions called facets, each with simple grey-level and shape constraints. The grey tones must be a polynomial function of the row-column coordinates of the pixels in the facet. Usually the polynomial will be of degree zero, one or two: representing constant facets, sloping plane facets and quadratic surface facets, respectively.

The edge detector based on this model estimates the underlying grey tone intensity surface at each pixel to determine whether it contains an edge. A least-squares fit with a functional form consisting of a linear combination of tensor products of discrete orthonormal polynomials is used to derive directional derivatives. A zero crossing of the second directional derivative indicates an edge.

Edge detection in itself was found to be inadequate for properly defining the grains. Experiment showed the need for small region elimination, achieved by symmetrically expanding surrounding regions, and sometimes an expand and shrink cycle was also applied to the edge pixels in order to close small gaps in the grain boundaries. This differed from a conventional dilation-erosion cycle in that a *conditional* erosion was used, which preserved the connectivity of the dilated image.

Following segmentation, the region attributes were computed and the regions, which could now be thought of as mineral grains, were classified / identified. GIPSY provided as many as thirty measurements for each segment in an image, allowing a property list to be compiled for each grain. Although GIPSY allowed identification on the basis of size, shape, surface texture or colour from multi-band images, mineralogical identification was generally made based on reflectance (grey-level) values.

An initial training phase was required, using ore samples of known reflectances to generate a segmented image which was labelled by a (human) operator using an interactive display. A simple decision rule, usually based on the average grey level intensities of the segments, was then obtained and used to identify grains in unknown samples. The classification procedure relied upon the differences in reflectance to separate the phases. Such separation became difficult if the reflectance differences were small and/or variable. Careful control of the wavelengths of the illuminating light was sometimes required to allow adequate discrimination.

2.3.4.1 Non-optical approaches

The problems associated with segmentation and labelling petrographic images have encouraged the development of radically different techniques involving non-optical image data.

One early example was Jones' adaptation of an electron-probe X-ray microanalyser so that it could be used as a linear image analyser [Jones 1982, 1984]. This instrument discriminates minerals by measuring the characteristic X-rays which are produced when small areas of the mineral are bombarded with an electron beam. Such systems are, generally speaking, too slow to be used to scan a whole image 'pixel by pixel'. Also, a system using only the microprobe may fail to detect the interfaces between adjacent grains of identical type. To overcome these problems, X-ray analysers can be used in conjunction with other systems based on either optical or electron microscopy. An advanced analysis system of this type has been described by Petruk [1988]: the Kontron image analysis system.

The Kontron system interfaces a microprobe, an energy dispersive X-ray analyser (EDXA), and an image analyser, with communication in both directions between the units. The analytical procedure involves transferring a backscatter electron image (BSE) from the microprobe to the image analyser. The minerals or phases are identified on the basis of their grey levels in the image. If the grey levels of two or more minerals or phases are too close for discrimination and identification, the minerals or phases are identified by scanning each grain with the electron beam of the microprobe under control of the image analyser, and analysing the grains with the EDXA.

To perform the image analysis a binary image is produced for each mineral displayed in the BSE image and prepared for analysis using a variety of image analysis routines. Thus the image analysis techniques are similar those used in systems such as GIAPP [2.3.3.2]. The binary images are analysed and the data are classified, summarised and output in simple tables and graphs.

Such non-optical techniques offer direct, accurate identification of minerals, circumventing both the segmentation and labelling problems. They are, however, costly because of the specialised hardware required.

2.3.5 Summary of findings

The above review of published work in the application of computer image processing and analysis to rock micrographs has demonstrated that computer-based techniques have much to contribute in petrographic and petrological studies and in engineering applications such as, for example, process mineralogy or reservoir studies. The use of computer image analysis has relieved the geologist from much of the routine drudgery involved in quantitative analyses, and has also provided new techniques which could not otherwise be achieved.

It would appear, from surveys of published work undertaken at the beginning and conclusion of this study, that few new image analysis techniques have been developed recently, although existing techniques continue to be widely used. The most noticeable advance appears to be in the use of non-optical methods for phase identification, in conjunction with conventional image analysis. This development reflects the fact that major obstacles to the wider application of petrographic image analysis lie in the segmentation and labelling of the image, i.e

(1) it can be very difficult to automatically segment the micrograph image (i.e. delimit the grains), and

(2) it can be very difficult to automatically label (i.e. identify) the grains in the segmented image, relying solely on simple optical information.

For these reasons segmentation and labelling of optical micrograph images still normally require a high level of interaction with the user.

The Petrophysics and Stone Conservation group at the University of Oviedo, which has been active in petrographic image analysis for over ten years, reports that the automatic discrimination of components in rock micrographs remains a central research problem [Montoto 1990; Menendez 1991].

2.4 Image understanding in petrography: a proposal

Considering the findings of the above survey, the following questions arise:

(1) How can image processing techniques contribute more to the solution of the segmentation and labelling problems?

(2) How can image processing techniques contribute more to petrographic studies in general?

It is suggested that a possible answer to both questions lies in the use of image processing routines which incorporate more of the capabilities of the human geologist. The geologist (henceforth, without loss of generality, masculine) has expertise on many levels. Consider, for example, only some of his abilities in the examination of a micrograph: he can segment the image, even if the minerals present are unknown to him; he can understand the three-dimensional structures which the two-dimensional cross-sections represent, such as the nature of the spatial relations between components (enclosure, intergrowth, twinning, etc.); he can probably recognise most of the minerals present; he can understand the geological significance of the inferred three-dimensional assemblage.

The expertise required for these tasks has various forms. Segmentation is usually performed with unconscious ease; only in some special cases, such as the presence of a twinned or zoned crystal, is some higher level knowledge required to achieve the correct interpretation. Similarly, an understanding of the three-dimensional structures represented in the section image is usually achieved with little or no conscious effort. The underlying mechanisms involved are, however, not apparent from casual introspection.

The other capabilities mentioned involve more explicit knowledge, acquired by formal training. For example, for mineral recognition the novice geologist must learn how to identify the key features in the mineral image, and match these against the features of minerals which have been learned or are looked up. Similarly, he learns to describe textural features and how to use this information in conjunction with his knowledge of the mineralogy in order to classify the rock, or make some other petrological inferences.

It should be expected that the higher level human expertise, which uses explicit rules, would be amenable to modelling via a rule-based intelligent knowledge based system (IKBS). IKBSs attempt to model the problem solving performance of human specialists in particular domains, using encoded human knowledge. Production rules [Davis *et al.* 1977] are the most widely used scheme for representing human knowledge in an IKBS. They are basically inference rules which can be presented in a simple notation as:

IF <premises> THEN <conclusion>

On evaluation of such a production rule, if the premises evaluate to true (i.e. if the logical combination of conditional statements evaluates to true) then the conclusion is assigned the value true and is added to the list of known facts. (The term expert system may also be used for an IKBS which embodies expert knowledge.)

IKBSs have been used in the way described. IKBSs exist for the identification of minerals in thin section micrographs [West 1985, Dagger 1990]. Dorans [1988] has implemented an IKBS for metamorphic petrography which uses production rules, and succeeds in capturing much of the geologist's expert knowledge and reasoning.

Thus the feasibility of constructing a system which incorporates higher level geological expertise has been demonstrated. In contrast, however, no work has been found reported which investigates the less formal and more intuitive reasoning abilities used by a geologist in the examination of a micrograph. (The expert systems mentioned above operate in a dialogue with the user, and derive all their information about the rock or mineral from him: the computer sees only through the user's eyes.)

It is suggested that the non-expert, 'lower level' image analysis capabilities of the geologist also warrant investigation. For example, the following questions may be considered: Is it possible for an image processing / analysis scheme to capture the features in a micrograph image which the human uses to segment, to understand spatial relationships, and to classify components? Can an automatic system produce descriptions of a rock thin section in terms meaningful to a geologist? If the answers to such questions are 'yes' then the key problems stated above will be nearer solution.

The next stage of this investigation therefore is a closer, more focussed examination of the features of crystalline rock sections, undertaken with these points in mind. Crystalline rocks have been selected for investigation since they are the subject of the more complex petrological studies, requiring phase identification and an understanding of rock textures. Mineral ores are also predominantly crystalline.

This investigation is the subject of the following chapter, which concludes with more detailed research proposals.

CHAPTER 3

Crystalline Fabric Analysis

3.1 Introduction

The term 'fabric', as used in petrology, embraces the terms 'texture' and 'structure': the fabric is the sum total of all geometric properties of a mineral aggregate. This chapter aims to show why (cross-sections of) crystalline rocks show some of the features they do, what the significant features are, and how these may be analysed and described.

Some details of igneous and metamorphic petrology are given, in order to explain some features of crystal shapes and crystal interrelationships. In particular, the 'geometry of overgrowth', ubiquitous in igneous rocks and ore assemblages, is explained and described. A discussion of the shape and symmetry descriptors which are applicable to individual crystals (crystal profiles) in polycrystalline rock sections is included.

The fundamental problem of 'boundary assignment' is considered in detail: the interface between two crystals is usually characteristic of one crystal only, and 'imposed' upon the second. An understanding of such relations between a crystal and its neighbours is required before meaningful descriptions of crystal shape and symmetry can be given. The identification of characteristic boundaries can also indicate the crystallisation sequence of minerals in a rock, and so provide petrological information.

The work of G. C. Amstutz and H. Giger is reviewed, since these authors have investigated how local features of crystal interfaces in sections can be used to infer important information about crystal relations. The features used by these authors for crystalline fabric analysis are, however, demonstrated to be inadequate for a full analysis of general polycrystalline rock sections.

A detailed investigation is then made of the indicators used by human observers to solve the problems of boundary assignment in simple crystalline fabrics. This problem has not previously been addressed in the literature. Explicit analysis rules are deduced, at first using an assumption of ideal crystal growth, and extended to accommodate non-ideal growth. The relative importance of the rules is also considered.

The chapter concludes with a statement of the various image analysis tasks which, it has been recognised, must be tackled if the boundary assignment problem is eventually to be implemented. A petrographic image analysis system based on these principles is outlined. This analysis motivates the work of the subsequent chapters.

3.2 Igneous rocks

3.2.1 Igneous petrogenesis

Igneous rocks form by the crystallisation (or glassification) of a molten magma. The nature of the resultant rock is governed by several factors. The bulk chemical composition of the magma is the basic control on which minerals can be formed in the crystallisation. The rate of cooling affects the size of the crystals which form and also has an influence on which minerals can form, as will be outlined below. If the melt is rapidly cooled, as is a volcanic lava flow reaching the surface, for example, the resultant rock will contain a high proportion of glass. With slower cooling, crystals form and the slower the cooling, the larger the crystals formed can be expected to be. The cooling rate basically depends on the depth of emplacement of the magma body, and its size. Thus there is a rough equivalence between classifications of rocks on the basis of their grain size (coarse, medium and fine grained) and the more genetic classification into deep-seated (plutonic) rocks, the rocks of minor intrusions (hypabyssal), and extrusive / volcanic rocks.

As mentioned, differences in the rate of cooling / crystallisation sometimes result in mineralogical as well as grain-size differences between rocks with the same bulk chemical composition. An explanation of the mechanisms operating is found in Bowen's *reaction principle* for the crystallisation of a silicate melt [Cox *et al.* 1974, p167]. This is not an inviolate rule, but an empirical principle based on commonly observed changes during the crystallisation of basaltic magmas. Bowen described two types of reactions between crystal phases and the liquid melt: continuous and discontinuous. In discontinuous reactions crystal phases formed at higher temperatures react with the remaining liquid at a lower temperature to form a new mineral phase or an altered version of the existing phase. In continuous reactions the solid and liquid phases form so-called solid solutions, with the crystals present gradually changing their composition. Both these types of reaction require an equilibrium between solid and liquid phases, which may easily be lost by over-rapid cooling. Interruption of continuous reactions in this way can lead to a change in the composition of crystals from core to margins, called zoning. Interruption of discontinuous reactions can preserve minerals which would otherwise have disappeared in later reactions. Different regimes of cooling applied to magma bodies of identical bulk chemical composition can therefore lead to the formation of rocks containing different minerals.

These principles go some way to explaining the fabric of an igneous rock. Crystals which form from the melt early, and which do not react with the melt, should be reasonably well formed, displaying characteristic faces. Crystals of the same phase forming close together at the same time may merge. Crystals formed later in the melt may grow against or around these solid islands and so compromise their form. This results in the 'fabric of overgrowth'. The formation of minerals in close contact, whereby the development of individual crystals is interfered with, and the whole locked together in a crystalline mass is termed 'paragenesis'.

A simple crystallisation model may be postulated in which there is no reaction between the phases which appear through time, and where each crystal phase appearing grows from an essentially random nucleation site and achieves its characteristic form 'as best it can'. That is to say if the growing crystal is too close to an existing crystal it may lose its ideal form as it accommodates the obstacle. This leads to a simple form of the fabric of overgrowth or occlusion, evident in a section of the resultant crystalline aggregate.

The same overgrowth textures are also found in ore assemblages, since this model again approximates the situation, with mineral crystals forming from a liquid medium. Sedimentary rocks may also show these textures in some situations when minerals have crystallised from the pore fluids.

The effect of viewing a section resulting from this simple crystallisation model can be described via an analogy. Suppose each crystal section is represented by a cardboard cutout in the shape and size which that crystal would attain if it grew without interference. The cardboard shapes are placed one at a time on a flat surface, in random positions representing arbitrary nucleation sites. Now view the result as a purely two-dimensional structure: cut-out profiles last to be placed will be seen in their true, unobscured shape; profiles placed early on may be partially occluded by later shapes, and may even appear as several disjoint patches or be wholly hidden. The analogy with the rock section should be clear, for the uppermost profiles can be seen to correspond to the earliest crystals to form, and the occluded shapes correspond to the later-forming crystals which have compromised their form against existing solid phases. The order of appearance (i.e. placing) of each profile is the reverse of the corresponding order of crystallisation.

Figure 3.1 illustrates this geometry of overgrowth in a schematic rock section. The numbers in each crystal profile represent the order of crystallisation. The numbering of the profiles is the sequence of crystallisation which this image represents, although this does not represent a unique solution: the labels of regions 1, 2 and 3 could be permuted and still give a perfectly valid ordering, for example. Regions such as 5a and 5b are parts of the same crystal, which can be assumed to join up in the third dimension.

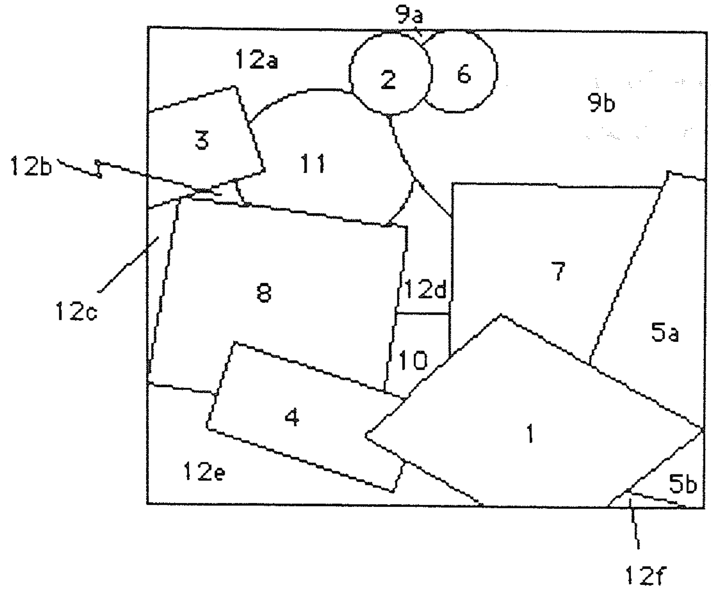


Figure 3.1 Synthetic section image

A consideration of sections such as figure 3.1 may suggest that in many cases it should be possible to deduce the order of crystallisation from an examination of the final solid state in the thin section. In some situations the crystallisation sequence may not be fully determinable, but again it should be possible to recognise these limits to interpretation. The various clues which can be used will be investigated below [3.5].

Looking ahead somewhat, it is notable that sometimes a relatively complicated set of clues will be required for analysis, which contrasts strikingly with the 'unconscious' ease with which the human observer can deduce the crystallisation sequence. The analogy described above suggests an explanation: people have highly developed skills for interpreting two-dimensional scenes of our three-dimensional world, with the analogous problems of occlusion, etc., and these skills translate naturally and directly to the petrographic domain.

3.2.2 Description of crystal form in igneous rocks

There are two aspects of crystal shape which it is important to specify: the quality of development of the faces of the crystal and the overall shape (in three dimensions).

3.2.2.1 Quality of development of faces

The terms most widely used to indicate the quality of development of the faces of a crystal are *euhedral, subhedral* and *anhedral.* These three terms are used to label the continuum of possibilities which arise in nature, from a perfect, complete set of characteristic crystal faces, to no (characteristic) faces at all: 'euhedral' means the crystal is completely bounded by its characteristic faces; 'subhedral' means the crystal is bounded by only some of its characteristic faces; 'anhedral' means the crystal lacks any of its characteristic faces.

Synonyms for euhedral are the terms idiomorphic or automorphic. Subhedral has synonyms hypidiomorphic or hypautomorphic. Anhedral has synonyms allotriomorphic or xenomorphic. This terminology may also be used in the description of ore assemblages.

It is important to note the word 'characteristic' in the above definitions. It is used to underline the fact that the crystal face must 'belong' to the crystal under consideration and not have been 'imposed' upon it by an adjacent crystal, as described above. Without the proper attribution of faces, clearly, the shape descriptions given above cannot be made, nor can symmetry descriptions.

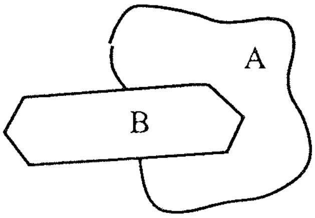


Figure 3.2 Euhedral and anhedral crystals

For example, figure 3.2 shows an anhedral, amorphous crystal of mineral A penetrated by one well-formed, euhedral crystal of mineral B. The crystal of mineral A appears bounded by its natural, irregular, shapeless border and the clearly defined crystal faces on its interface with the crystal of B. This illustrates how a mineral grain may appear to display some well defined crystal faces, and yet in fact be wholly anhedral and not subhedral.

One example is the so-called ophitic texture found in some gabbros. Here a very large augite crystal fills the interstices between and partially encloses a number of well-formed plagioclase crystals. In this situation, a thin section may show a patch of augite surrounded by plagioclase crystals and so entirely bounded by well-formed crystal faces. However the faces may be said to belong to or be characteristic of the plagioclase and are not characteristic of the augite. (It should also be noted that current opinion is that the two phases did crystallise together simultaneously - not plagioclase first and augite second, as would be expected - but at different rates of nucleation [Williams *et al.* 1982, p57].) This task may be complicated by the

In simple situations, such as the ophitic texture, the geologist has no difficulty in recognising the characteristic faces and correctly attributing them. Indeed this process is taken for granted as part of the observer's common sense, and is usually not even mentioned in introductory texts. However, as was implied above, correctly attributing crystal faces is not a straightforward image processing task.

Once the perimeter of a crystal section has been analysed to find the characteristic faces, describing the quality of development of faces follows easily. The terms euhedral, subhedral and anhedral could, for example, be replaced by a 'hedrality index', say, given as the fraction of the perimeter which consists of the crystal's characteristic faces.

It may be observed that if the simple crystallisation model described above is assumed for igneous rocks and ore bodies then the task of determining the characteristic faces of crystals is found to be equivalent to determining the relative order of appearance of the crystals. This fact makes petrological studies such as those of Amstutz (described below [3.5.1]) relevant for consideration.

Another interesting image analysis possibility arises: it may be feasible to use a knowledge of the usual sequences of crystallisation to aid identification of the crystals in an igneous rocks, by comparison with the sequence deduced from inspection of the overgrowth fabrics. Similar studies may be used to help differentiate between igneous and metamorphic rocks (see below).

3.2.2.2 Crystal habit

The second aspect of crystal shape which must be considered is the overall shape in three-dimensions. Just as existing crystals in the melt interfere with the growth of characteristic faces in later appearing crystals, they will also constrict their overall growth. In some situations it may be important to describe the actual, compromised form, with all its imposed concavities and inclusions. Often it is the 'ideal', uncompromised form which is of interest, for mineral identification, for example.

Different sets of faces on crystals may be differently developed, giving rise to various crystal 'habits'. For example, crystals may be needle-like (acicular), blade-like, platy (very flattened in one direction), tabular (slightly flattened in one direction), botryoidal (forming rounded masses with very few rational faces), and so on. Certain habits or forms may be characteristic of some minerals and so may be used as a guide to identification. Determining the form of a mineral viewed in a thin section may require the examination of several examples of cross-sections before an idea of the three-dimensional form can be made. This task may be complicated by the different appearance of different sections of anisotropic crystals.

A useful description of crystal form may therefore be derived from the lengths and symmetry relations of the characteristic faces. The symmetry of crystals will be considered separately below [3.4].

3.2.3 Crystal shape: convexity and non-convexity

Although wide variations of crystal shape are possible, the ideal form of nearly all common minerals is a convex solid. (A two or three dimensional shape is said to be convex if a straight line segment drawn between any two points in the shape lies completely within the shape: thus, for example, a circular disc is convex, and a circular annulus is not.) In reality, however, crystals will often fail to achieve perfect convexity.

A major cause of non-convexity is growth against existing solid crystals in the melt, as has been described. However, even with no such physical barriers to growth, local differences in conditions in the melt may cause growth to proceed at different rates for different parts of the crystal, which may result in local concavities. Where crystals grow contemporaneously at close nucleation sites their growths may interfere with each other, and this may easily result in non-convex crystals. Crystals may also be partially resorbed by reaction with the liquid, which may result in a rounding off of crystal profiles and the formation of embayments in the profiles - although embayments may sometimes be simply the result of 'non-ideal' crystal growth, as described above.

Extreme examples of non-convex growth (without resorption) are found in so-called skeletal crystals, which have hollows and gaps, possibly regularly developed. In thin section the spaces appear as embayments and holes in the crystal, filled with groundmass crystals or glass, and the crystal may appear as isolated patches. The patches can be identified as parts of a single crystal because they share optical and crystallographic characteristics. Parallel growth is one form of skeletal growth producing a crystal which in thin section may appear as a bundle of apparently separate elongated crystals whose axes are mutually parallel, or nearly so. These phenomena are illustrated in the atlas of MacKenzie *et al.* [1982; pp.20-22].

Twinning may also produce non-convex crystals. As described above [2.2.5], twins are single crystals in which different parts of the lattice are differently oriented. One simple example of a non-convex twin is the swallow-tail twin: imagine a parallelopiped reflected in one face.

3.3 Metamorphic rocks

3.3.1 Metamorphic petrogenesis

Metamorphic rocks are fundamentally different from igneous rocks in that they are not formed by the crystallisation of a melt but by essentially solid state reactions. These metamorphic transformations occur as a rock adjusts in order to achieve a more stable combination and arrangement of minerals. The instability is the result of a new regime of temperature, pressure and chemical conditions being imposed upon the rock. Three principal types of metamorphism have been recognised on the basis of field criteria. *Dislocation* metamorphism occurs along zones of intense deformation such as faults. *Contact* metamorphism occurs in areas adjacent to igneous bodies, as a result of the intense heating. *Regional* metamorphism occurs over extensive areas and is not clearly related to any local event.

Metamorphic changes typically include some or all of the following: the crystallisation of new minerals not previously present (neomineralisation), the loss of some original minerals, the development of 'new' crystals with undistorted shapes in place of 'old' distorted and strained crystals of the same mineral (recrystallization), and the formation of new textures and structures. The new textures may be produced by the deformation of existing structures or the new crystals formed adopting particular arrangements during their growth.

Thus, although some metamorphic rock textures may sometimes resemble igneous textures their significance is different. Crystals have not grown in an enclosing, homogeneous melt but in a solid medium. The physical properties of crystalline solids, especially those relating to speed of growth and stability of boundaries, vary from one mineral to another and from one direction to another within an individual crystal. These differences are largely responsible for the textural details of metamorphic fabric. Similar factors do play a part in forming igneous fabrics but are overshadowed by the much greater physical differences between crystalline solids in general and the melt phase within which these solids have developed [Williams *et al.* 1982, p438].

3.3.2 Description of crystal form in metamorphic rocks

The terminology for describing crystal shape in metamorphic rocks is similar to the igneous nomenclature. A grain of a metamorphic mineral bounded by its own crystal faces is termed *idioblastic*; a shapeless crystal grain is termed *xenoblastic*. The term *crystalloblastic* is applied to textures and structures resulting from the growth of crystals in a solid medium.

F. Becke (see the paper by Kretz [1966]), the founder of modern metamorphic petrography, realised that certain minerals have a greater ability to display crystal faces than others. Accordingly, he arranged the common minerals of metamorphic rocks into a hierarchy of groups, termed the *crystalloblastic series*, of which the following is a portion:

....
{sphene, magnetite,..}
{garnet, kyanite, staurolite..}
{pyroxene, hornblende}
{albite, dolomite..}
{muscovite, biotite...}
{calcite}
{quartz, plagioclase}
{orthoclase, microcline}
....

Any mineral in this series is supposed to be capable of displaying its form against all minerals in the groups below it. This ability or tendency was termed 'Kristallizationskraft' by Becke, later translated to 'force of crystallisation'. This term is often replaced by 'form energy', to avoid confusion with the phenomenon exhibited when a crystal exerts a force during growth. Becke himself, and many others, have noted exceptions to the rule associated with the crystalloblastic series. Kretz [1966] suggested the concept may be refined by using instead a hierarchy of mineral interfaces, arranged in order of increasing specific interfacial free energy, e.g.:

...
garnet (110) - quartz
biotite (001) - quartz
hornblende (110) - quartz
...

- where the bracketed numbers indicate the Miller indices of the particular faces (this indexing system is explained in any introductory crystallography text, for example that of Phillips [1971] or Cox *et al.* [1974]).

One consequence of the factors described by Kretz is that crystal A, say, may impose its form on crystal B at one local interface but crystal B may impose its form on crystal A at another interface. This would introduce some complications into the use of the established crystalloblastic series hierarchy as a guide in the identification of mineral grains in metamorphic rocks, along the lines of the scheme suggested for igneous rocks.

The other scheme mentioned above, for discriminating between igneous and metamorphic fabrics, would depend upon the identification of relative impositions of crystal form and a comparison of the sequence found with the crystalloblastic series, and the common sequences of crystallisation in igneous rocks. Such a scheme should also exploit the rules for discriminating igneous and metamorphic textures given below.

3.3.3 Metamorphic textures c.f. igneous textures

Metamorphic rocks commonly show textural and structural features which are much rarer in igneous rocks. These features result from the crystal growth in a solid medium, and so may be used to distinguish crystalloblastic fabrics from the fabrics of igneous rocks [Cox *et al.* 1974, p212]. Briefly, they are:

a) Metamorphic rocks have a much larger proportion of crystals showing little or no crystal form.

b) Minerals which commonly occur as crystals of inequidimensional shape often have these forms exaggerated in metamorphic rocks in comparison with igneous rocks. Inequidimensional grains tend to lie in definite orientations in metamorphic rocks, giving rise to a foliation.

c) The larger crystals in metamorphic rocks ('porphyroblasts') frequently contain inclusions of small mineral grains, and these inclusions are often arranged in some regular manner.

d) Compositional zoning and twinning of mineral grains are much less common in metamorphic rocks than in igneous.

3.4 Crystal symmetry description

Only when the characteristic faces and boundaries of a crystal have been recognised can an appraisal of the symmetry be made, since the principal indicators of the crystal symmetry are the rational or characteristic faces (non-characteristic faces cannot be used). Crystal cleavages can also provide clues about internal symmetries and structure since they represent the natural planes of weakness in the crystal.

It must be stressed that it is the internal symmetry of the crystal which is of primary interest, and this will very rarely manifest itself in a crystal which is symmetrical in the normal, everyday sense of the word. In general the different crystal faces will be developed to different degrees, producing a crystal shape lacking any obvious symmetry.

Before considering the symmetry of two-dimensional crystal sections the symmetry of crystals in three dimensions should be understood. Only a brief treatment is presented here: full details may be found in the texts of Phillips [1971] or Cox *et al.* [1974], for example. It is convenient to first consider regular crystals, with equal development of faces. Their shapes may be described with reference to symmetry planes, axes, and a centre of symmetry.

A symmetry plane or mirror plane is an imaginary plane which divides the crystal in two pieces which are mirror images of each other. A symmetry axis is a line about which the crystal may be rotated so that it achieves a position of congruence (i.e. it looks the same as it did originally) at some stage during the rotation. If congruence is achieved after 180 degrees, for example, the axis of rotation is a 2-fold symmetry axis or diad axis. Only a small number of symmetry axes are possible in natural crystals: the diad, the triad (3-fold), the tetrad (4-fold) and the hexad (6-fold). A centre of symmetry exists when every face of the crystal is matched by one parallel to it on the other side of the crystal.

Crystals can be divided into seven groups, known as crystal systems, on the basis of the symmetry axes present. The systems and their characteristic symmetries are as follows:-

cubic	4 triad axes
tetragonal	1 tetrad axis
hexagonal	1 hexad axis
trigonal	1 triad axis
orthorhombic	3 diad axes
monoclinic	1 diad axis
triclinic	no axes

A step nearer the real world can now be taken and the existence of crystals whose irregular development disguises their symmetry considered. The angular relationships between the crystal faces remain constant regardless of the degree of their development, so these angular relationships become the focus of interest. It is usual to represent each face by an outward face normal, which is a vector normal to the face, pointing outwards. The angular relationships between face normals may then be considered. In practice a stereographic projection is used to translate this problem into that of studying the relationships between points on a two-dimensional stereogram of the face normals.

The problem of studying the symmetry of a two-dimensional crystal profile can be transformed in a similar way. The simplifying assumption may be made that each straight line face represents a plane face in three dimensions which is normal to the plane of the section. The stereogram then reduces to a circle (the equatorial great circle of the full stereogram) and each face normal is represented by the point on this circle where the tangent has the orientation of the crystal face: figure 3.3 gives an illustration.

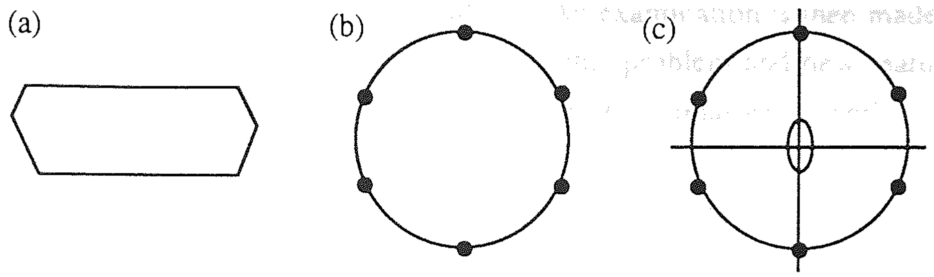


Figure 3.3 A crystal, its sketch stereogram, and symmetry analysis

This simplified stereogram summarises the important orientation information in a useful form; the symmetries of the pattern of points on the stereogram may now be examined in order to reveal the relevant symmetry properties. Figure 3.3(c) shows the results of the analysis for the stereogram of figure 3.3(b): there are two perpendicular mirror planes indicated by lines, and a diad axis indicated by the central symbol. (These properties place the crystal in the orthorhombic group.) It may be noted that the original crystal section of figure 3.3(a), because of the non-regular development of faces, is outwardly asymmetric.

3.5 The identification of characteristic boundaries

The problem of correctly assigning faces and boundaries to crystals has been identified above as being of fundamental importance in analysing a crystalline fabric. It is central to crystal description and mineral identification since it is impossible to describe the symmetry of a crystal or the quality of development of its faces until one has identified which of the boundaries of the crystal are in fact characteristic of the mineral, rather than characteristic of adjacent crystals which appeared earlier in time and so imposed their form upon their later-forming neighbour. The assignment of boundaries problem may sometimes be referred to here as 'determining crystal precedence' since, as a general rule, establishing the ownership of interfaces between crystals is equivalent to determining their relative order of crystallisation.

It appears that the problem of identifying characteristic faces or boundaries has not been addressed in the literature; the importance of characteristic boundaries is mentioned, but no observations on how they are to be recognised have been found. The task appears to be treated as one which may be solved by the microscopist without explicit guidance. The closely related problem of identifying the relative order of crystallisation from examination of a section has been addressed by Amstutz and Giger, who have reported two indicators of crystal precedence for ore analysis: interface convexity and what are termed 'false corners'. These analysis rules are described below and, by example, it is shown that they are inadequate for a complete solution of the boundary assignment problem. An examination is then made of the various features which are important in solving this problem and new features are identified. Explicit analysis rules are derived and their relative 'power' discussed. Simple crystalline fabrics are considered, where the assumption of perfect growth is made; non-ideal growth is considered in section 3.5.3, and the analysis rules are found to be still relevant. Section 3.5.4 presents a summary of the analysis principles derived. Section 3.6 looks at the implications of these findings for further research.

3.5.1 Amstutz and Giger's quantitative fabric analysis

Amstutz and Giger have made a special study of intergrowth fabrics and the development of quantitative analysis techniques, with special reference to ore analysis [Amstutz & Giger 1970, 1972]. Fabric analysis is important in ore studies on two counts. Firstly, it is valuable to recognise and report the intergrowth patterns of different ore minerals, in order to estimate the grinding which will be required for their liberation, and to monitor the actual results. (This has already been described [2.3]; further discussion may be found, for example, in the papers of Petruk [1982] or Barbery [1984]). Secondly, as described above [3.2.1], it has long been appreciated that a study of rock fabrics can yield useful genetic information. Until the work of Amstutz, that information had been only qualitative in nature and not quantitative.

Amstutz and Giger provided an historical overview of rock fabric studies in their two papers [1970, 1972]. They noted that qualitative descriptions have been used since last century, but "...it seems that in the field of geometric studies of mineral aggregates there has been an actual standstill since ... 1910" [Amstutz & Giger 1972]. They correctly foresaw, however, that the development of quantitative techniques such as stereology, coupled with the use of electronic scanning devices and computers, could bring new possibilities to all fields dealing with physical and genetic properties of rock aggregates.

One of the first contributions to the quantitative study of paragenetic relations in ore minerals is due to Amstutz [1965]. This paper indicates that a paragenetic sequence for ore bodies can be derived using an objective, quantitative treatment of intergrowth relations, derived from two simple principles: the superposition rule and the concave-convex principle. The superposition rule, applied to minerals crystallising from a fluid and forming in veins, states that phases were first deposited on the wallrock, and later phases formed on top of the earlier phases. The concave-convex principle is related in that it assumes that earlier formed crystal phases are almost invariably 'nuclei' for later superposed grains. Consequently, earlier phases must have proportionately more convex surfaces against later phases, and *vice versa*.

(The only conditions where this principle has to be reversed is the filling of spherical cavities.)

Amstutz demonstrated that measurement of the concave-convex statistics in crushed ore samples allowed the study of the overall paragenetic sequence of the parent mineral deposit: examining sizeable, complete sections was no longer necessary. Amstutz and Giger [1970, 1972] continued the development of quantitative analysis, introducing stereologic principles to express the qualitative relationships between minerals in a reasonably quantitative way. In particular, the 1970 paper improved the definition of the index of intergrowth or locking used previously and introduced the idea of 'straight intersection numbers'.

The (improved) locking or intergrowth index for two phases A and B is given by the quotient: the surface density of the intergrowth surface between A and B divided by the average surface density within the sample. The surface density is the surface-to-volume ratio for a given phase; the surface density of the interface between phases A and B is meaningful since the interface is a surface in three dimensions, and is the area per unit volume of the interface. More details of such stereological parameters may be found in the standard text of Underwood [1970].

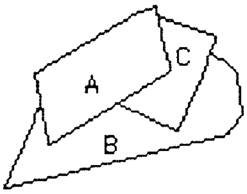


Figure 3.4 'False corners'

The straight intersection number (or 'Durchgangszahl') is a topological parameter which may be used as quantitative support for qualitative suggestions on time-sequence or paragenetic relations in fabrics. It is related to the concave-convex test described above. The crystallisation processes in mineral aggregates are processes in time, and mostly discontinuous (in time). As was established above, this means that the earliest generation of grains is surrounded by the next one: a sequence reflected in the geometry of overgrowth. Figure 3.4 illustrates this relation: the boundary of the earlier grain A is straight or at least smooth at the point where the interface between two subsequent grains (B and C) terminates against it. These sorts of triple points in sections can easily be distinguished (by eye) from those 'true' triple points produced where three grains grow together. These 'false' triple points have been termed 'false corners'. If one determines all the false corners of grain type A produced by grains B, one obtains the mean straight intersection number of B given A:

$$I(A|B) = \frac{N(A|B)}{N(A)}$$

where N(A|B) is the total number of all straight intersections phase B produces with the investigated grains of phase A, the number of which is N(A).

3.5.2 Establishing crystal 'precedence'

It is important to note that the techniques developed by Amstutz and Giger were designed for a *statistical* analysis of the intergrowth relations in a rock sample. In this section it is shown that the indicators described above, namely false corners and convexity of crystal interfaces, are insufficient to determine automatically in all situations the correct paragenetic relations between a given crystal and its neighbours. Further indicators of crystal precedence are presented: the use of the assumption of underlying convex form is shown to require in some situations a stronger test than the simple examination of interface curvature, and the role of parallelism and other symmetry features is recognised. The relative importance and 'power' of these indicators is also considered.

3.5.2.1 Crystal inclusion and convexity relations

As has already been described, in the crystallisation of an igneous rock later-forming crystals may form against or around existing, earlier-formed crystals. Since crystals tend to a natural convex form this process is the basis of Amstutz's concave-convex principle: the convexity of an inter-crystal interface reveals the relative precedence. When one crystal region in a section completely encloses another this provides a reliable and robust indication of crystal precedence: no measurement of interface convexity need be made since the fact of inclusion is clear. When a cluster of crystals is enclosed this fact indicates that all the crystals of the cluster precede the enclosing crystal but gives no information about precedence relations within the cluster.

When one crystal is only partially surrounded by another the interface convexity can still reveal the relative precedence, as Amstutz observed. Figure 3.5 shows some possibilities: figure 3.5(a) shows complete enclosure; figures 3.5(b) through (d) show partial enclosure; figure 3.5(e) shows no enclosure and no relative precedence can be deduced. Region adjacency is sometimes quantified by expressing interface length as a proportion of region perimeter [Nazif 1983; Barrow & Popplestone 1971]. It may be noted that this measure cannot be used to derive precedence information, except in the case of total enclosure: interface convexity is significant where adjacency alone is not.

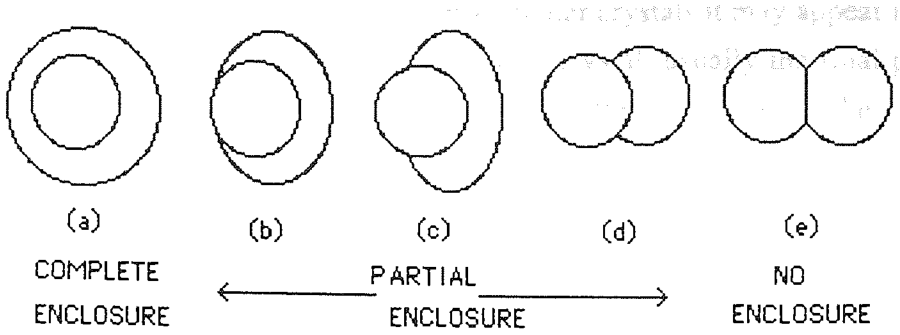


Figure 3.5 Inclusion styles

There are also situations where partial inclusion may occur between two crystals with a straight interface. Figure 3.6 gives one example: crystal A partly surrounds crystal B but intervening crystals reduce the interface to one straight line segment.

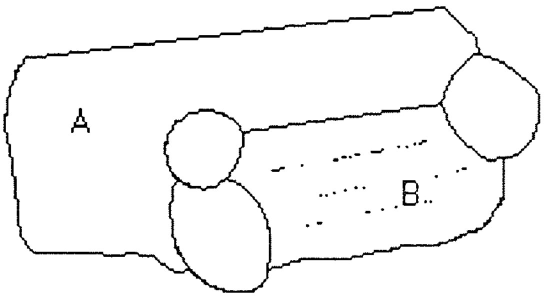


Figure 3.6 Partial enclosure with a straight interface

In such situations the partial enclosure relationship between the crystals can be revealed by consideration of the convex hulls of the two crystal regions. The convex hull of a shape S, which will be denoted H(S), is the smallest convex set of points which contains S; convex shapes are their own convex hulls. Applying these ideas to figure 3.6 it can be observed that H(B) and A do not intersect, whereas B shares approximately half of its area with H(A). These intersection properties reflect the fact that A partially surrounds B, and so identify the correct precedence.

Thus when two crystals share a straight interface the intersections of their convex hulls can reveal a precedence which the Amstutz concave-convex test cannot. Although the convex hull test is more powerful than the interface test the latter should still be retained since it would often be sufficient to reveal precedence and it would be simpler to implement.

It may be noted that if, for two crystals A and B, both H(A) and B and H(B) and A have non-empty intersections, then this suggests a more complicated intergrowth style. This situation can only arise when crystals violate the convexity assumption being maintained at this stage, and the matter will be considered again below [3.5.3].

When one large crystal surrounds many smaller crystals it may appear in cross-sections as many isolated patches, with the large crystal, usually the final phase to crystallise, filling the interstices between the other crystals. The correct understanding of this type of texture requires that the isolated patches of crystal be recognised as portions of one single crystal (not simply crystals of the same type). Once this has been established, consideration of the convex hulls will again indicate the true precedence relation. Recognition of the separated regions as portions of the same crystal may be difficult in practice. The regions must show the same, or very similar, colours in all orientations in plane polarized light both with and without the analyser, and their textures and cleavage patterns must be consistent.

3.5.2.2 False corners

The second indicator of precedence used by Amstutz was the presence of 'false corners'. The basis for this test is again the principle that later-forming crystals may form around or against existing crystals. The use of false corners to infer crystal precedence is, however, limited since the configuration may arise purely by chance. Similarly, a false corner may by chance appear like a true triple point.

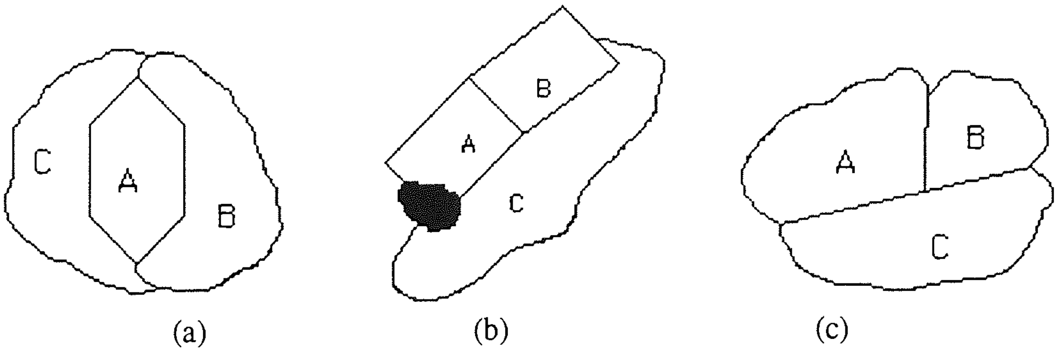


Figure 3.7 Illustration of the limitations of the 'false corner' heuristic

Figure 3.7(a) provides an example where crystal A precedes both its neighbours B and C yet the vertex where the three regions meet does not give a clear indication of the situation (precedence in this figure can, however, be deduced from convexity of the AB, AC and BC interfaces). Figure 3.7(b) is an example of a situation with a misleading vertex. Here, as revealed by convexity, crystals A and B both precede crystal C, yet the ABC vertex resembles a false corner which would suggest that C precedes both A and B. Figure 3.7(c), however, illustrates one situation where the false corner test is the only method which indicates the true relationship between the crystals, i.e. provides the interpretation which a geologist would make, namely that C preceded both A and B..

Thus it has been shown that false corners are relatively 'weak' indicators of precedence, in that the evidence of convexity, for example, can overturn the conclusion which would be made on the basis of vertex configurations alone.

3.5.2.3 Symmetry

Figure 3.8 provides examples of rock section configurations where a geologist would be able to infer the assignment of boundaries but where the indicators of Amstutz described above cannot provide the answer. These situations show the importance of symmetry.

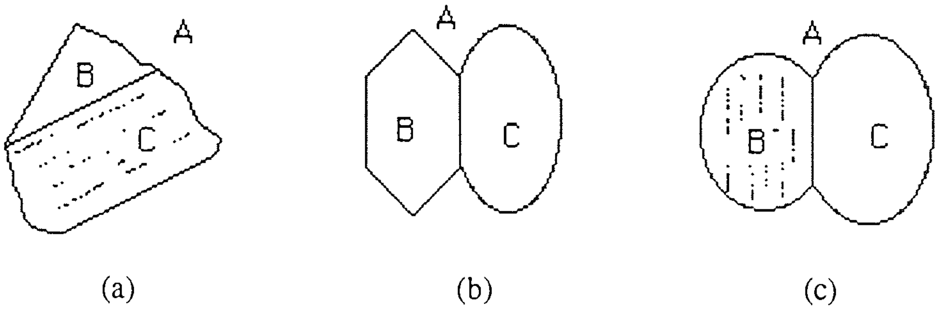


Figure 3.8 Interfaces assigned via symmetry

Figures 3.8(a), (b) and (c) show crystals labelled B and C surrounded by crystal A. Consideration of convexity or inclusion therefore provides the correct interpretation in all three examples that B and C both precede A. In all cases, however, the BC interface is perfectly straight and the convex hulls H(B) and H(C) do not intersect. Convexity therefore gives no information on the relative precedence of B and C. The ABC vertices do not show false corners and again support no inferences. Nevertheless, positive inferences can be made.

In figure 3.8(a) the geologist can perceive that the BC interface is characteristic of C alone and that C precedes B. This is because the BC interface is parallel to the opposite face of crystal C (which is clearly characteristic of C, because of convexity and inclusion considerations). As additional evidence for this interpretation the BC interface is seen to be parallel to the cleavage of crystal C. Thus the BC face fits the pattern of symmetry associated with crystal C. There are no appropriate symmetry properties which associate the interface with crystal B. It is therefore natural to assign the interface to C.

Figure 3.8(b) illustrates a similar situation: convexity and false corners tell nothing of the precedence between crystals B and C, yet the BC interface can be seen to fit a symmetric pattern if assigned to crystal B as a characteristic boundary. The interface yields no such pattern if assigned to crystal C. The interface can therefore safely be assumed to be characteristic of crystal B. In figure 3.8(c) amorphous crystals B and C have an interface which is parallel to the cleavage of B. Crystal C shows no cleavage. In the absence of other evidence it is therefore likely that the interface is characteristic of B.

Thus it would appear that the human observer of a rock section can decide whether a certain face is characteristic of a crystal by seeing if it fits into some existing recognisable 'pattern', or gives rise to a recognisable pattern. For example, if the acceptance of a putative rational face is found to result in, say, a regular, symmetric crystal, it is virtually certain that this face *is* characteristic of the crystal. As described above [3.4], the ideas of crystal symmetry are slightly different from 'conventional' symmetry: it is the underlying symmetry, revealed by the angular relations of faces and cleavages, which is important and a symmetric crystal may appear to be non-symmetric according to the normal use of the term. In practice the commonest symmetry feature is likely to be parallelism between faces or parallelism between a face and a cleavage direction (since cleavages can be thought of as 'failed faces').

3.5.3 Accommodating non-convexity

The previous sections have shown how it is often possible to establish crystal precedence relations in polycrystalline sections when the crystals concerned have convex ideal form. In this section the question of how the analysis rules derived above extend to non-convex crystal growth will be considered. It will be shown that the rôles of symmetry and false corners remain unchanged, and rules for deriving precedence based on convexity can be modified to accommodate non convex growth.

Three forms of non-convex growth have been described above [3.2.3]: irregular development, mutual intergrowth and the formation of true non-convex crystals. In irregular development the ideal crystal form is convex but this is not perfectly attained due to local variations in crystallisation conditions. An extreme form of this is skeletal growth. Such growth still carries implications about crystal precedence and the ownership of interfaces. Intergrowth implies simultaneous growth and a shared ownership of the interface. True non-convex crystal growth is very rare in igneous rocks.

Twinning may produce non-convex crystal but the twin components themselves are usually convex. The twin may be recognised as a cluster of (conventional) crystal regions showing certain geometric and symmetry features [2.2.5].

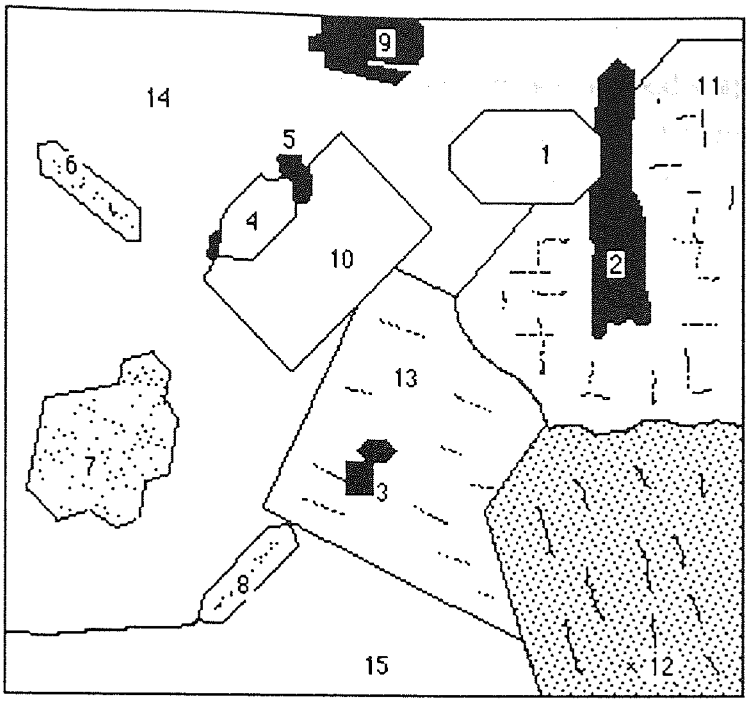


Figure 3.9 Olivine-malignite micrograph image

Figure 3.9 reproduces a drawing of a thin section micrograph from a standard petrological text; the igneous rock represented is an olivine-malignite [Hatch *et al.* 1972, p280]. The process of drawing clearly involves simplifications, but this may still be considered a realistic image in terms of the structures represented. In this section many crystals do not show perfect convexity (even ignoring occlusion effects), yet the assignment of boundaries is relatively clear.

Region 2 shows slightly irregular form, yet the 2:11 and 2:14 interfaces are clearly characteristic, since region 2 is nearly convex and is partially enclosed by regions 11 and 14. (The 1:2 interface is assigned to region 1 on the grounds of convexity, symmetry and false corners.) The 9:14 interface is assigned to region 9 since the region is again partially enclosed. Further, the interface shows faces with consistent symmetry properties: the faces seen within the concavity are parallel to other faces on the interface.

The 12:13 interface is clearly assigned to region 12 on the grounds of convexity, a false corner and symmetry - one face is parallel to the cleavage. The 11:13 and 11:12 interfaces show irregular form with no symmetries linking them to other features. There is no overall convexity on the interfaces: the regions seem to penetrate each other to a similar extent. A likely conclusion for both interfaces therefore is simultaneous growth. It may be noted that this then implies simultaneous growth for regions 12 and 13, although their interface could be unambiguously assigned to region 12. In this case factors such as stability of boundaries and nucleation rates (described above in the consideration of the ophitic texture and metamorphic crystal interfaces) must have acted to produce an interface characteristic of one region only.

Regions 9 and 12 in figure 3.9 show the importance of good shape in boundary assignment, and the role of faces and symmetry. The faces of region 9 form two groups of good symmetry, suggesting two contiguous components to the crystal; the good faces on region 12, one showing a false corner and lying parallel to the cleavage, identify the characteristic nature of the 12:15 and 12:13 interfaces. If a face has been identified then it may be possible to assign it to a region on the grounds of symmetries between it and other elements. Once a face on an interface has been assigned to a crystal the entire interface can be assigned with it. Conversely, if an interface is assigned to to a crystal, perhaps on the grounds of convexity or the symmetry of one particular subface, then all faces on the interface are assigned. These may then serve as sure characteristic features against which faces of uncertain ownership can be compared to see if their symmetry relations give grounds for assignment.

One other important feature of crystal profiles which can help decide the ownership of boundaries is the 'character' of an interface, in particular its smoothness. A smooth, undulate interface such as the 11:13 interface of figure 3.10 is a good indication of intergrowth. A jagged interface tends to suggest irregular growth and may indicate an interface characteristic of one region only. Smoothness appears to be inadequate in itself as a basis for assignment of faces but can provide useful supporting evidence.

These examples show that the precedence cues derived previously [3.5.2] are still useful in analysing sections where crystals need not have ideal convex form. However, the convexity tests suggested above require slight modification to be applicable in these new circumstances.

The suggested analysis was first to check if the interface was curved, since a convex interface in the ideal situation could be assumed to be an unequivocal indicator of precedence. For straight interfaces the intersection properties of the convex hulls would be considered.

The convex hulls of regions A and B may be denoted H(A) and H(B). Abusing the notation by making region labels also represent the areas of the regions, the area of intersection of each region with the hull of the other is found as a fraction of the total area. This would form two quotients for neighbouring regions A and B: $Q_{AB} = (H(A) \cap B) / B$, and $Q_{BA} = (H(B) \cap A) / A$. These quotients indicate the proportion of the crystal surrounded or partially surrounded by its neighbour.

If the crystals had grown perfectly, i.e. with no concavities other than those resulting from growth around or against pre-existing crystals, then one of these quotients would be zero and the other would indicate the proportion surrounded by its neighbour. Calculating the two quotients for the neighbouring regions 2 and 11 in figure 3.9 gives $Q_{2,11} \approx 0.02$ and $Q_{11,2} \approx 0.9$. Thus one quotient is not quite zero. The two values reveal how region 2 is surrounded by region 11, and indicate minor concavities on the interface.

Using such quotients for the 11:13 interface might, however, give misleading results. The two areas of intersection are approximately equal but the actual region areas are not. Thus the numerators of the quotients are similar but the denominators are dissimilar, leading to a difference in the quotient values and a spurious implication of surroundedness. It therefore seems more appropriate to directly compare the areas of intersection: in general terms, H(A)∩B and H(B)∩A.

If both these areas are small or zero there would be no implication of partial enclosure and one could also deduce that the interface must be straight or nearly so. (The size of areas could be judged by comparing them with the length of the interface.) If both areas were non-zero then their ratio should reveal the likely precedence: if the area H(A)∩B is very much greater than H(B)∩A then B precedes A and the interface is characteristic of B. If H(B)∩A is zero then convex growth is implied; if H(B)∩A is non-zero then some irregular growth is implied. If both H(A)∩B and H(B)∩A are large and of comparable size then an intergrowth is likely.

This form of the test on region and convex hull intersections therefore allows partial enclosure to be deduced in cases of irregular growth, and recognises possible cases of intergrowth. In many cases, such as the adjacent crystal pairs 11 and 12 or 11 and 13 in figure 3.9, this modified form of the analysis is equivalent to forming the convex hull of the interface, partitioning the hull by the original interface and then comparing the two areas. Figure 3.10 illustrates this process.

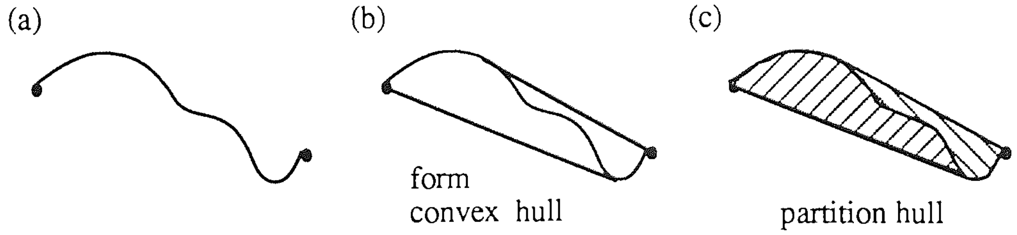


Figure 3.10 Analysis of an interface using the convex hull

However, the two tests are not the same, since some crystals may partially intervene between a pair of adjacent crystals so reducing the extent of the interface. In such situations this simpler form of the test, since it is only concerned with the interface, could miss implications of partial enclosure. The measure which considers convex hulls of the entire region is therefore seen to be preferable. If both areas of intersection are large however then it may be useful to consider in addition the partition of the interface in order to obtain more information about how the regions penetrate each other.

3.5.4 Precedence indicators: summary of rules

The concave-convex test on crystal interfaces and the false corner test for vertices reported by Amstutz have been examined. Limitations on the use of false corners have been demonstrated. A stronger test on convexity has been proposed, applicable in situations where the interface is straight or where the regions have no direct interface. This test has been extended to support inferences in situations when non-convex crystal growth has occurred. New precedence indicators based on considerations of crystal symmetry have also been presented.
Explicitly, the indicators of crystal precedence and characteristic boundaries are:

1) Enclosure
If crystal B wholly surrounds crystal A then crystal A may be assumed to have crystallised first and the AB interface may be assumed to be characteristic only of crystal A.

2) Interface convexity
If the interface between crystals A and B is convex, with A relatively convex and B relatively concave, then the AB interface may be assumed to be characteristic only of crystal A.

3) Partial enclosure
If the convex hull of a crystal region A, i.e. H(A), intersects a crystal region B, and H(B) has no intersection with A then A at least partially surrounds B and so B may be assumed to precede A, and any portion of their interface may be assumed to be characteristic of B. This test may be applied when the interface is a simple straight line, which by itself carries no relative convexity information. The implications on relative precedence resulting from this test do not require the regions concerned to have a direct interface

Rules 2 and 3 apply when convex ideal crystal growth can be assumed. Both rules can be viewed as special cases of the following rule which is applicable even with imperfect convex growth.

4) Partial enclosure (stronger form)
If the area of intersection of region A with the convex hull of region B is very much greater than the area of intersection of region B with the convex hull of region A, i.e. $A \cap H(B) >> B \cap H(A)$ then A may be assumed to precede B, and any portion of their interface may be assumed to be characteristic of A.

5) Symmetry

If the interface between crystals A and B, or a face on that interface, fits into a recognisable (and mineralogically relevant) symmetry pattern with other elements of crystal A, and does not do so with elements of crystal B, then the interface may be assumed to be characteristic of A

Corollary 1: A face on a crystal profile which is parallel to a known characteristic face of the crystal is likely to be characteristic.

Corollary 2: A face on a crystal profile which is parallel to the cleavage of the crystal is likely to be characteristic.

6) False corners

If, at the point where three crystal regions A, B and C meet, the boundary of C is straight or at least smooth where the AB interface terminates against it then C can be assumed to precede both A and B, and the BC and AC boundaries can be assumed to be characteristic of C. The limitations on this test have been described above: the inference may be overturned by stronger evidence.

3.5.4.1 Rules in combination

In resolving the ownership of some crystal boundaries more than one of the rules presented above may be applicable, or one rule may be applicable at several points on an interface. It is possible in such situations that the various local inferences might not produce a consistent interpretation: the results of one test may contradict those of another. Figure 3.7(b) provided one example, where the false corner test was misleading. It therefore seems that the rules have different 'powers'. It is considered that the ordering of rules given above in fact reflects this ranking.

The strongest precedence indicators appear to be enclosure and partial enclosure. Complete enclosure (rule 1) gives a robust indication of crystal precedence, regardless of the convexity of the enclosed crystal. When ideal convex growth is assumed, partial enclosure is revealed by rules 2 and 3 above; when possibly irregular growth is acknowledged, partial enclosure is revealed by rule 4. These inferences cannot be refuted and any contradictory evidence, such as a false corner or face-cleavage parallelism, must be attributed to chance.

Parallelism or, more generally, good symmetry (rule 5) appears to be a stronger indication of precedence than false corners (rule 6). Where two rival symmetries give contradictory interpretations then the higher symmetry is likely to be the better guide. These principles simply take the most probable interpretation, since a configuration of higher symmetry is less likely to have arisen purely by chance.

These considerations suggest that recognising the ownership of faces and interfaces in a polycrystalline section image can be a relatively complex pattern recognition task, involving many separate judgements. Introspection might not suggest that the human observer uses local tests on vertices or interfaces, but the suggestion here is that this must be part of the process: this is how he or she appraises the overall shape properties of the profiles and identifies the elements which fit some pattern.

This matter has been considered in greater detail. Strong parallels have been found between the principles derived above for solving the assignment of boundaries problem and principles of perceptual grouping. A report of the investigations has been presented and published elsewhere [Thomson & Claridge 1989]. Some implications of the linkage between polycrystalline section analysis and more general vision tasks will be discussed below [Chapter 8].

3.6 Implications for subsequent work

As outlined in this chapter, the analysis of polycrystalline rock sections is an important task for geologists. An essential prerequisite for such analysis is the assignment of 'ownership' of boundaries between adjacent crystals. Section 3.5.2 above has presented the features of polycrystalline sections which reveal the ownership, based on an assumption of ideal convex crystal growth; section 3.5.3 has examined how these principles extend to cases of non-convex growth. Section 3.5.4 has presented a summary of the rules for boundary assignment which have been derived, and described their relative 'power'.

Example rock section configurations have demonstrated the usefulness of the rules in determining characteristic crystal boundaries and establishing relative precedence. Although these rules may not be sufficient to capture the performance of an expert geologist in the analysis of polycrystalline sections, their value here is that they clarify the nature of the image analysis tasks which need to be tackled if the boundary assignment problem is eventually to be automated. In particular:

(a) The recognition of partial enclosure of one crystal region by another requires the development of methods for the calculation and manipulation of convex hulls.

(b) Methods for crystal symmetry analysis must be developed. Since the elements which reveal symmetry are crystal faces and cleavages, this requires that

(c) A method for identifying faces or possible faces on a crystal profile must be found, and

(d) Techniques for recognising cleavage cracks amongst other textural features must be developed. Further,

(e) An implementation of the T-junction test for false corners is required.

(f) Methods are needed to implement the extraction, representation and storage of the information which is input to and output from the various steps in solving the overall problem.

(g) A mechanism and a strategy for controlling the operation of the separate analyses are also required.

Subsequent chapters investigate how these requirements may be met. Chapters 4 through 7 address tasks (f), (c), (b) and (g) respectively. The recognition and analysis of crystal cleavage, the implementation of the T-junction test for false corner analysis, and the development of methods for finding and manipulating convex hulls (i.e. tasks (d), (e) and (a)) have been considered but are not treated in detail here. Investigations of crystal cleavage analysis and convexity analysis have been reported elsewhere [Thomson & Sokolowska 1988; Thomson 1991]; chapter 7 summarises the findings.

Figure 3.11 outlines a possible control structure where a supervisor module acts as an interface between the different analyses modules and the database of spatial information, controlling the operation of the modules, supplying their input data, and updating the database with the new information produced by the analyses. At the completion of the analysis the database should contain the necessary information regarding the assignment of characteristic boundaries, and other useful descriptive information about the crystals.

Chapter 4 develops a system which extracts the required spatial information from a pre-processed image where the basic crystal regions have been found and stores this information in a database structure. The input image, where regions which are uniform in the sense of their features have been found but the characteristic boundaries have not been identified, may be described as 'partially segmented' [Levine 1985, p386].

Chapter 5 develops a new algorithm for multiscale contour analysis, suitable for finding possible faces on a crystal profile. It is shown that, in general, symmetry must also be considered in making the final selection of faces.

Chapter 6 considers symmetry analyses which are appropriate for completing the process of finding crystal faces begun in chapter 5, and for assigning the ownership of interfaces.

Chapter 7 reviews the level of implementation attained for the various component analyses and proposes a control strategy for the overall task.

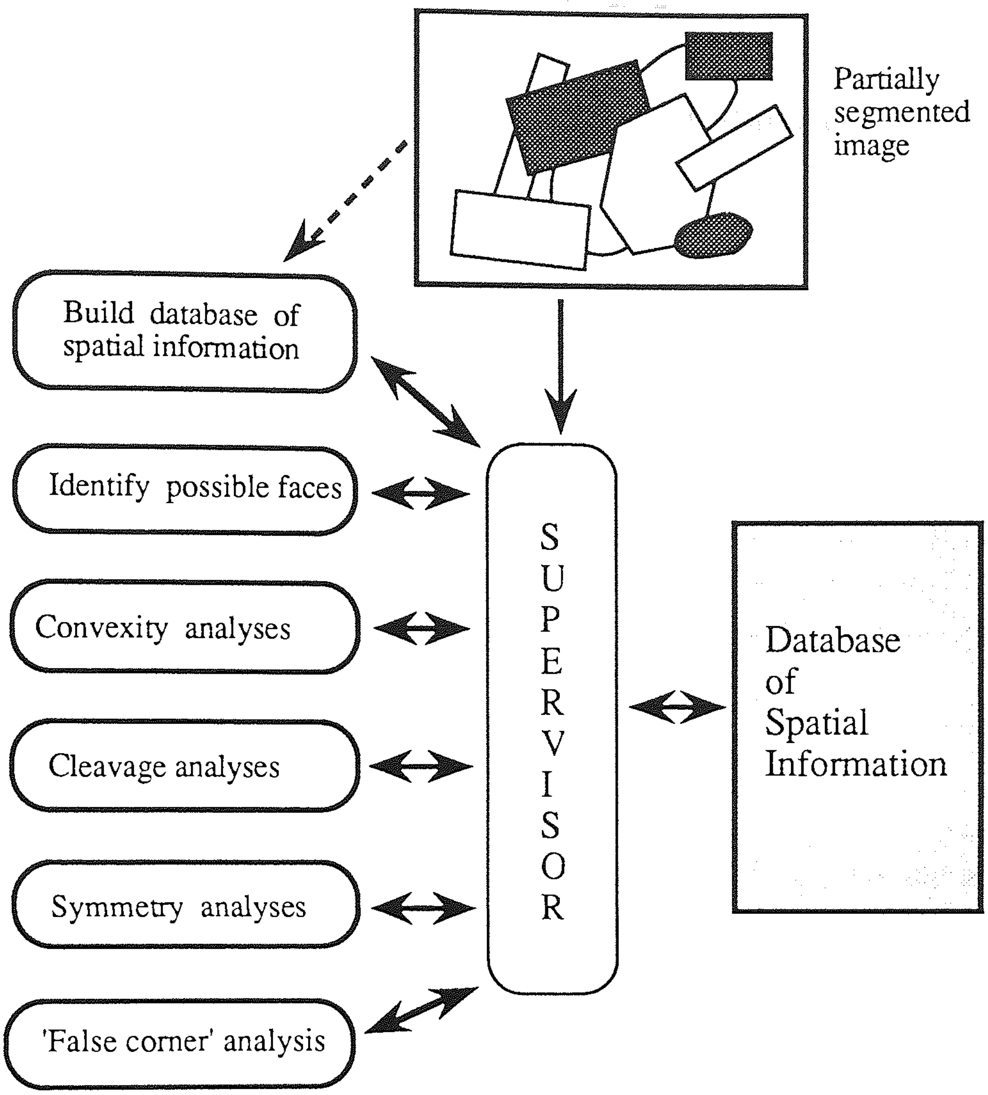


Figure 3.11 Petrographic image analysis system outline

CHAPTER 4

A Spatial Information Database

4.1 Introduction

The extraction of spatial information from a partially segmented image and the representation of this information in symbolic form in a database are fundamental to the implementation of the image analysis scheme outlined above [3.6]. Such a database will provide the input data for analysis routines; the results of the analyses will be added to the database. At the conclusion of this programme the database can be examined to produce descriptions of crystal shapes, inter-crystal relations, etc., and so may act as a source of information for higher level analyses.

This chapter considers the creation of the database: the extraction and storage of a set of basic descriptions for the 'primitives' in the image, such as regions, boundary lines and vertices. These two aspects, extraction and representation, are dealt with separately. Subsequent chapters will consider algorithms for the analysis of properties such as convexity, symmetry, and the presence of crystal faces; the image features which are considered here are much less domain specific. The structural analysis and database construction algorithms may therefore find wider application.

Short reviews of relevant work are presented [4.2.2, 4.3.1]. In spite of the widespread need for the derivation and representation of basic information from partially segmented images no complete solution to the problem at hand was found. The algorithms subsequently devised to meet these ends will be described in detail. First, as a necessary prerequisite to such discussion, the basic properties and definitions of discrete images are briefly considered.

4.1.1 Digital connectivity

Dealing with a discrete space necessitates some revision of normal geometric definitions associated with continua. Detailed treatments of this subject have been given, for example, by Rosenfeld and Kak [1982; Ch.11], Rosenfeld and Melter [1989]; this section concentrates on definitions of connectedness and their implications.

A point (i.e. pixel) P of a digital image may be specified by two integer coordinates (x,y). Assuming it is not on the edge of the image, P has two horizontal and two vertical neighbours:

(x-1,y), (x+1,y), (x,y-1), (x,y+1)

These are called the 4-neighbours of P, and they are said to be 4-adjacent to P. P has four diagonal neighbours, namely

(x-1,y-1), (x-1,y+1), (x+1,y-1), (x+1,y+1)

These, together with the 4-neighbours, are called the 8-neighbours of P, and are 8-adjacent to P. A path from point P to point Q in the image is a sequence of points P = $P_0, P_1, \ldots P_N$ = Q, such that P_J is a neighbour of P_{J+1}. If 4-adjacency is used in the definition of neighbour then the path may be called a 4-path, and similarly for an 8-path. A subset of the image, i.e. some collection of pixels, may be called *connected* if a path can be found between any two pixels in the set. Again, this provides the definitions of 4-connected and 8-connected regions.

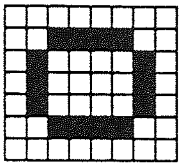


Figure 4.1 Example of the digital connectivity paradox

Figure 4.1 illustrates a possible paradox arising from these definitions. If 4-connectedness is assumed then the marked (black) pixels must represent four regions; if 8-connectedness is assumed then the marked pixels represent one connected region, but its interior is connected to its exterior. The problem is resolved by using 8-connectedness for the marked pixels, and 4-connectedness for the unmarked pixels: this gives one connected region with an interior not connected to its exterior. Connected sets of pixels which do not have any such 'holes' in them are called *simply connected.*

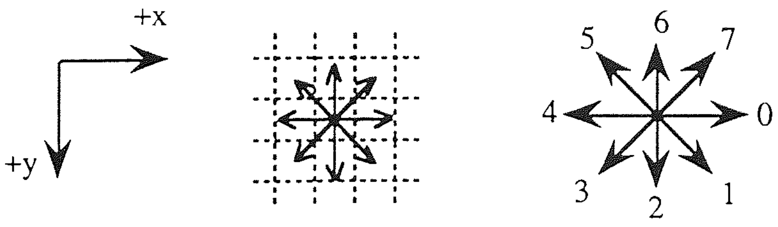


Figure 4.2 Freeman chain coding

When a continuous curve is represented on a uniform rectangular grid the local direction is restricted to one of eight directions, assuming 8-connectedness, or four directions for four-connected curves. These local directions can be represented by code numbers, using only two or three bits. This is the simple idea of Freeman chain coding [Freeman 1970]. The chain code provides a compact description of the curve, in a form suitable for further analysis. Figure 4.2 illustrates the coding system used in this study; the left-handed axes may be noted.

4.2 Image spatial information and its representation

Two basic issues will be considered: (1) what are the features of a crystal profiles image which must be described, and (2) how can this information be represented in a form suitable for manipulation and interrogation by analysis routines.

In accordance with the discussion above [4.3] it will be assumed that the input image has been partially segmented, so that units such as individual crystal profiles or twin components have been identified as individual regions. The regions are 4-connected and separated by borders which are 8-connected. The image may therefore be represented using only two pixel values, say 1 for a region pixel and 0 for a border pixel. This can also be thought of as a line drawing in which all lines begin and end at junctions (vertices) or are closed. Following a taxonomy of images given by Freeman [1978], such an image will be termed a 'region map'.

4.2.1 Spatial information from crystal profiles images

The important features in crystal profiles image have been identified above, in chapters 2 to 4. These features will be re-examined, to find the implications for the extraction and representation of spatial information. There are three obvious primitives in a crystal profiles image: regions, borders and vertices. To what extent is it necessary to catalogue these primitives, to describe them and their interrelationships?

The analysis of the shape of a crystal profile will require the extraction of each region contour in the image. This can be conveniently stored in a chaincoded form. A possible alternative is to store a region representation, for example a hierarchical decomposition such as the quad-tree [Samet 1980], rather than the bounding contour. The contour representation is considered more suitable in view of the subsequent analyses. It also permits the extraction of specific interfaces, i.e. the portion of the perimeter of grain A shared with grain B, so that features such as concavity or convexity, roughness or smoothness may be examined. Some analyses may require the area of a region; this can be derived from the chain code [Freeman 1974].

It is necessary to recognise vertices in an image i.e. points where three or more regions meet. Incidence information relating vertices to regions and boundaries is also required. 'False corner' analysis clearly requires a knowledge of the vertex-region incidence. Vertices also provide natural points linking the descriptions of neighbouring regions.

Information regarding adjacency between regions is also important. As has been described above, evidence from adjacent crystals may be required before characteristic boundaries can be identified. Adjacency information has been seen to be central to petrological studies [2.3.3.1]. A special case of adjacency information which should be extracted concerns regions adjacent to (i.e. cut off by) the picture frame. These regions need to be recognised so that they are not used in shape analysis and the frame is not confused with a crystal face.

Inclusion is a special form of adjacency which may usefully be made explicit. Information regarding the number of inclusions in a region and which regions include others is useful in the assignment of characteristic boundaries and the identification of crystals.

4.2.2 Representing spatial information: some examples

Some published designs for the representation of two-dimensional spatial information will now be discussed, and also an unpublished data structure developed in the early stages of this study. This is not a comprehensive review of the area; it is not intended as such. Nevertheless, it is considered that enough key ideas are introduced and alternative methods presented to give a perspective on the design adopted for implementation [4.2.3].

4.2.2.1 The 'segmented image database'

Oddy *et al.* [1983] described aspects of the software design of an image processing facility. This facility included a spatial information system to provide a database for handling segmented images, for example in high level pattern recognition tasks. The organisation of their Segmented Image Database (SID) was based on the CODASYL database design [Martin 1977]. Two basic data structures, data records and indices, were used to construct a data network, using pointers. The database was implemented entirely in FORTRAN. Various aspects of the database design were discussed in the paper; no information was given on how the spatial information was extracted from the image.

The image 'primitives' used in SID were regions, holes, boundaries, and line segments. Each primitive occurring in the image was given a unique index (label), and a data record describing it was placed in the database. Vertices were not treated as special primitives, but information about vertices could be deduced since line segments were defined as terminating at vertices. Line segment records contained the coordinates of the beginning and end of the chain code (beginning and end vertices), the chain code length, the chain code itself, and pointers to the records of the regions to the right and left of the line segment. The definition of line segment used guaranteed that there could be at most one region on either side of the line segment.

Each region record had associated with it information regarding its holes, adjacent regions, boundary segments, and region attributes such as the minimum and maximum x and y coordinates occurring in the region. This was achieved by the region record containing pointers to a series of indices - a hole index, adjacency index, etc. - and these indices held pointers to the relevant records. The adjacency index associated with each region record stored pointers to regions which were adjacent or touching. The boundary segment record denoted the boundary of a region and held pointers to the records of the line segment which made up the boundary.

The SID database provided a structure which gave access to attributed region and vertex adjacency graphs, allowed searching for regions with a given attribute, and allowed access to all attributes and other parameters for a given region, including the provision of a chain coded representation of the region boundary for shape analysis. Many features of SID were used in the network database described below.

4.2.2.2 The 'patchery data structure'

Ohta [1985] presented a complete system for knowledge-based interpretation of outdoor natural colour scenes. The regions obtained as a result of segmentation were recorded as a two-dimensional image array. A symbolic description of the image was then required to arrange the segmentation results into a well-organised data structure which allowed easy derivation of pictorial features related to the properties of and relations between the regions. The data structure was called the 'Patchery Data Structure'.

The descriptive elements (primitives) in the Patchery Data Structure were regions, boundary segments, vertices, holes and line segments. In Ohta's terminology a 'line segment' meant a *linear* segment of a boundary segment, and a 'hole' meant a group of regions surrounded by another region.

Each descriptive element had associated with it a set of primary features. The term 'primary' meant that these were features which had to be derived directly from the image data. These features were stored in records associated with each descriptive element found in the image. Secondary features, which could be derived from primary features, were not stored in this way, but calculated when needed.

The primary features of a vertex, for example, were its position and number of boundary segments incident; the primary features of a boundary segment were chain code, length, and contrast (a measure of difference in intensity of regions on either side); the primary features of a line segment were length, positions of end points, the line orientation, and distance from origin. Only the contour length was stored as a primary feature of each hole; regions had ten primary features associated.

The topological relations among regions, boundary segments, vertices, holes and line segments were expressed by pointers between each descriptive element record. The pointers used in the Patchery Data Structure are illustrated in figure 4.3. The boundary segments which form the contour of a region or a hole are ordered counter-clockwise.

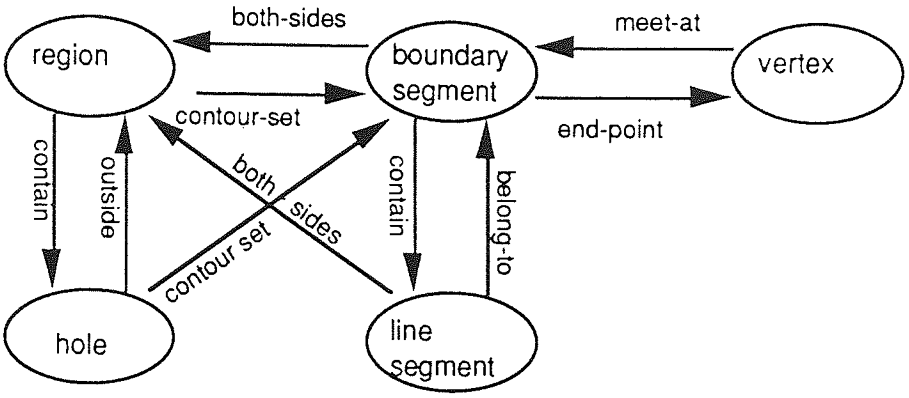


Figure 4.3 The patchery data structure

Following the principle of storing only primary features, only the 'primary relations' were explicitly represented. Other relations could be derived from the primary ones when they became necessary. For example, the set of regions touching a certain region could be obtained as follows: (1) using the 'contour-set' relation, the set of boundary segments surrounding the region was derived; (2) using the 'both sides' relation, regions which shared the boundary segments with that region were gathered as the objective set.

Features of the patchery data structure were used in the developed network data structure [4.2.2.3] and similar relations were used in the SPIN database [4.2.3]. A straight linear segment primitive was not used in the basic form of the database, although later analyses could add such information.

4.2.2.3 A network data structure

In the early stages of this study a data structure was implemented for use in the automation of routine petrographic image analysis tasks, such as those performed by GIAPP or GIPSY [2.3.3.2]. The data structure was primarily designed to hold information describing region adjacency, interface descriptions (as chain codes) and vertex configurations for, respectively, crystal adjacency studies [2.3.3.1], concavity-convexity and 'false corner' analysis [3.5.2.2].

The data structure was designed as a 'supergraph' of the region adjacency and vertex adjacency graphs. The nodes of vertex adjacency graph (VAG) represent the vertices in an image; graph edges link nodes whose corresponding vertices are connected by an interface, i.e. the vertices are neighbours on the boundary of some region. The basic image primitives used were regions, vertices, interface segments and holes. For each occurrence of a primitive in the image a record of appropriate type was created and linked into the structure. Pointers from record to record represented the relations between the corresponding primitives.

Region records held the region label, a pointer to a separate region property table which held location, size, shape information etc., the number of external neighbours, the number of holes within the region, and other pointers required for establishing adjacencies which will be described below.

Vertex records held a unique label, the vertex coordinates, the degree of the vertex (i.e. number of neighbouring regions), pointers to the records of the incident interfaces, pointers to the records of the incident regions, and also a pointer which provided a direct linkage between vertex records and so allowed quick traversal of all vertex information.

Interfaces were curve segments. Their records held a unique label, start and end coordinates, pointers to the records of the terminal vertices, curve length and a chain code representation. The identity of the regions on either side of the interface were found via the incidence records of the terminating vertices. 'Pseudovertices' were introduced on interfaces without true vertices, as a computational convenience. Complete region contours were considered 'secondary' items and could be assembled from the relevant set of interfaces if required.

As in the patchery data structure holes were regions or groups of regions wholly contained within a single region. As is explained below [4.3.1.2], in this analysis holes were considered to be the basic components of the image, since they corresponded to disjoint components of the vertex adjacency graph. The profiles image itself was viewed as a hole in the exterior region (the 'picture frame'). A hole record held a unique label, a pointer to the record of the surrounding region, and a pointer to the first vertex in the hole component.

The data structure was implemented in C. This programming language allows dynamic allocation of arrays, used for storing chain codes, and provides a pointer data type which allowed direct implementation of the 'conceptual' pointers in the network. The use of pointers was complicated by the fact that many 1-to-N relationships have to be modelled, and the degree of the relations (the value of N) could not be known in advance. For example, a region could have an indefinitely large number of external neighbours, and so representing region adjacancies required a region record to have a pointer to a linked list of pointers to the neighbouring region records. A similar arrangement was required for the holes within a region, and for the regions on the exterior of a hole. In some of these situations additional 'summary' records were introduced to hold information which could provide shortcuts when accessing data items.

Further details of this network structure will not be given here since, although implemented, the design was considered to have features which made it unsuitable for use in analyses more complex than those for which it was designed. It could prove difficult to extend the structure to incorporate new items of information such as symmetry, polygonalisations, etc. The extraction of data items from the structure could also be awkward to implement, although fast in operation, since customised access routines had to be written for each.

It proved extremely difficult to update the structure with new information; often the simplest way of updating the complex network was to competely rebuild. (In its originally intended use updating would not have been required.) It may also be noted that this structure was practically impossible to store in a file. A simpler, more flexible approach was required for the sorts of applications described above [4.4].

4.2.2.4 The McGill relational database

The McGill University rule-based expert system for low level image segmentation [Nazif 1983] has been described briefly above [4.2.2]. Within this system a 'short term memory' (STM) stores image-specific data. Its contents can be accessed and modified by a suite of processing modules. The STM is implemented as a relational database [Codd 1970].

The basic storage unit is a relation which is defined on a finite number of domains D1...Dn. Information is stored as a set of n-tuples (d1,..,dn) such that di belongs to Di. Each relation has a set of key domains which uniquely identify the tuples. Database entries are found by searches on the key field values. A relation may be thought of as a table of n columns with each row representing one n-tuple. The tables of the STM database store descriptions of lines, regions and their spatial interrelations.

There are three tables for region description. Each region has a unique label which is used as the key value to access the information held in the remainder of the tuple. One table stores region properties such as area, perimeter, circularity, the defining coordinates and aspect ratio of a minimum bounding rectangle. A second table stores the chain code of the region boundary. A third table stores region adjacency information. For each region a tuple stores the number of adjacent regions, N, and N pairs of values representing the labels of the adjacent regions and the corresponding adjacency value [3.5.2.1]. These latter two tables appear to violate the defining principles of a relational database since the tuples in the tables can have variable length.

There are five tables for line description. As with regions, each line is given a unique identifier which is the key value for database access. The first line data table stores end point coordinates, length, and chaincode description. The second stores a series of descriptors for line shape and contrast. The remaining tables catalogue relations between a line and its near neighbours in terms of lines which are behind, in front and parallel. A third series of tables describe line-region relations such as regions to the left of a line, lines which intersect a region, etc.

In general, the simplicity of this database organisation facilitates modification and allows a uniform data access mechanism to be used. Some slight loss of efficiency is associated with this simplicity of use since access involves searching the database tables. The relational database model can be seen to overcome the problems encountered in the network scheme. It was therefore used as a basis for the spatial information database finally implemented in this study.

4.2.3 The structure of the SPIN spatial information database

The SPIN database stores the key spatial information describing an image. Not all the possible information described above [4.2.1] is stored explicitly, but it is implicit, and can be deduced if required from an analysis of the stored data. The SPIN database is implemented as a POP-11 list structure, which is easily stored, retrieved, updated and analysed by POP-11 routines. This section describes the database organisation in detail; its construction is described below [4.3.2 ff].

The basic database structure is that of a relational database, with some modifications: in effect, some data tables are stored within tables in order to improve efficiency of data retrieval. The database consists of eight tables, each of which describes some aspect of the image, e.g. vertices, or region adjacencies. Each table consists of a set of 'tuples' or 'records' which describe one instance of the feature. Thus, for example, the outer contour table is a set of records each of which describes the outer contour of a region in the image.

The SPIN database is a list of elements; each element is a sublist: a database table. Each table is itself a list, of indeterminate length, of sublists: the tuples. In order that the tuples remain a known length and format, sublists are used as entries where the information refers to an indeterminate number of entities. This is in contrast to the scheme used in the McGill database. Details will be given below.

The first eight tables of the database represent the basic spatial information.

These tables are supplemented by later routines which add shape and symmetry information, for example. The eight tables are:

1. region table
2. outer contour table
3. hole contour table
4. vertex table
5. outer contour / vertex table
6. hole contour / vertex table
7. external neighbours table
8. internal neighbours table

(1) The records of the region table are 2-element lists:

```
region_tuple = [ region_label  [region_property_list] ]
```

The first element is the label of the region, which uniquely identifies it. The second element is a list which may be used to store the properties of a region generated by later processing, such as area, mean greyscale, circularity or elongation, for example.

(2) The records of the outer contour table each describe the external contour of one region in the image. The records are five-element lists:

```
outer_contour_tuple = [ region_label  xst  yst
                                 no_chainlets  [code] ]
```

`region_label` is the label of the region whose contour is being described. `xst` and `yst` are coordinates of an arbitrary position on the border of the region, the start-point for the chain coding. `no_chainlets` is the number of chainlets in the chain code; this code is stored in the final sublist.

(3) The records of the hole contour table are similar to those of the outer contour table:

```
hole_contour_tuple = [ region_label  hole_no  xst
                          yst  no_chainlets  [code] ]
```

`region_label` is the label of the region which contains the hole being described. The holes within a region are identified by numbers 1, 2, 3... and `hole_no` is the number of the hole under consideration. The other list entries are analogous to the `outer_contour_tuple` entries.

(4) The records of the vertex table each describe one vertex in the image. The records are five-element lists:

```
   vertex_tuple = [ vertex_label  xpos  ypos  order
                        [incident_regions_list] ]
```

`vertex_label` is a unique label identifying the vertex. `order` is the number of regions which meet at the vertex or, equivalently, the number of border segments which meet at the vertex. The labels of the regions incident are stored in the final sublist in the tuple, in no particular order. A sublist is used since the number of regions is variable.

(5) The records of the outer contour / vertex table contain the information which relates the vertices to the contour representations, giving the location of the vertices in terms of chainlet numbers of the contour chaincode. Since the number of vertices on a contour is indeterminate sublists are used to keep the records uniform:

```
outer_contour_vertex_tuple =
      [region_label no_vertices [vertex_contour_sublists]]
  vertex_contour_sublist  =  [vertex_label  chainlet_no]
```

`no_vertices` is the number of vertices lying on the contour. This number may be zero, in which case the list of `vertex_contour_sublists` will be empty. The list of `vertex_contour_sublists` has one entry per vertex on the contour, and each entry is a two-element list. The first element is the label of the vertex, the second is the number of chainlets one must track along the chaincode from the code origin in order to reach the vertex. The coordinates of the code origin are stored as `xst` and `yst` in the `outer_contour_tuple`. (It is possible for the same vertex to appear more than once on the contour of a region: vertex 16 appears twice on the contour of region 27 in figure 4.8).

(6) The records of the hole contour / vertex table are similar:

```
  hole_contour_vertex_tuple = [region_label hole_no
                       no_vertices [vertex_contour_sublists]]
```

There is one such tuple for every 'hole' in the image, including the whole image, which is seen as a hole in the exterior region. (This convention allows one to identify regions which may have been 'cropped' by the frame.) Two 'key' values are required to locate the hole's record in the table: the region label and the number of the hole within that region.

(7) The region adjacency information is stored in two tables, since a distinction is made between internal and external neighbours. The records of the external neighbours table are four-element lists, where two list elements are sublists:

```
region_external_region_tuple =
  [region_label no_ext_nbrs [region_labels] [vertex_pairs]]
```

no_ext_nbrs holds the number of external neighbours of the region with the label region_label. The labels of these neighbouring regions are stored in the sublist of region_labels. The second sublist holds in corresponding positions the labels for the pairs of vertices which define the interface between the region and its neighbours.

(8) The records of the internal neighbours table have an additional level of structure:

```
  region_internal_region_tuple =
            [ region_label  no_of_holes [hole_nbr_records] ]
  hole_nbr_record  =  [ hole_no  no_nbrs  [region_labels]
                                   [vertex_pairs] ]
```

This table records the adjacencies between a region and regions which are wholly or partly surrounded by it. Instead of embedding a database table (the hole_nbr_records list) within each tuple, an alternative scheme would be to use tuples with a double key (region_label and hole_no) as for the hole contour / vertex table entries.

An example of the SPIN database for a simple test image is given below [4.3.2.4].

4.3 The extraction of image spatial information

The extraction of the required topological and structural information from a partially segmented image [4.2.1] will now be considered, i.e. establishing procedures for deriving descriptions of the basic image elements - regions, borders and vertices - and their adjacency / incidence relations. An algorithm for efficiently deriving the information required for the construction of the SPIN database will be described. This is prefaced by a brief consideration of relevant work in this area, including a description of an unpublished analysis scheme based on graph traversal.

4.3.1 Structural feature extraction: some examples

The exact form of the algorithms required will clearly depend upon the format of the image. As described above, the partially segmented image to be analysed is assumed to be a region map, i.e. a collection of 4-connected regions of pixel value 1 separated by border pixels of value 0. Such a region map may be pre-processed in order to facilitate subsequent analysis. Two useful procedures are border thinning and region labelling.

Region labelling, also called component labelling or blob-colouring, assigns a unique label (integer value) to each region, physically changing the pixel values of the region to this label. This can be achieved in two row by row scans. On the first scan the pixel values of the upper and left hand 4-neighbours of non-zero pixels are examined. If both are zero (i.e. border pixels) then a new region label is created and the pixel set to this value. If only one neighbour is zero then the pixel is given the value of the non-zero neighbour. If both are non-zero and equal the pixel is given this value. If both are non-zero but unequal the pixel is given the smaller value, and the equivalence of these two values as region labels is recorded. At the end of this scan every region pixel has been altered to a region label, but many labels represent the same region. The equivalences must therefore be unravelled and unique, consecutive integers assigned to the equivalence classes. On a second pass over the image the region pixels are reset to these new labels.

Thinning reduces the inter-region borders to a minimal thickness which preserves connectivity. This enables the border segments to be Freeman chain-coded, and allows simplifying assumptions to be made when analysing the local structures of vertices and interfaces. Several 'standard' thinning algorithms have been established; the matter is considered, for example, by Rosenfeld and Kak [1982; sect.11].

Several books, notably the reference cited above, describe algorithms for region labelling, boundary and line tracking, and the extraction of region adjacency, surroundedness and nesting properties. However, no treatment has been found detailing an efficient scheme for the full analysis of region maps, combining these elements. Such schemes have been presented for the analysis of binary images, a simpler class of image where the *regions* may be labelled using only two pixel values, and there are no explicit borders.

A region map is a form of binary image, but the image analyses described would not be appropriate. Binary images have a simple adjacency relation between regions: if regions share a border then one encloses the other. This enclosure relation defines a tree structure. Milgram [1979] described algorithms for building these region trees, incorporating the generation of border chain codes which are stored at the tree nodes.

The algorithms used boundary tracking, where the region boundaries were pursued to closure in turn. Suzuki and Abe [1985] performed a similar analysis. They labelled region borders as internal or external and established a 'parent' relation which allowed the nesting of regions to be elicited. The notion of internal and external boundary following was used in the algorithm finally implemented [4.3.2].

4.3.1.1 Structural analysis in GIAPP

The GIAPP program package for the analysis of rock micrographs has been discussed above [2.3.3.2]. This system could be used to derive quantitative structural information using techniques of geometric probability and mathematical morphology. Fabbri *et al.* [1983] and Chung *et al.* [1984] have described how the basic adjacency information is derived and tabulated by the system.

The input is a hand digitised binary image: a region map. As described above, the borders are thinned and the regions are labelled. Interactive phase labelling is then performed, producing a second image in registration with the first, where the regions bear labels indicating the mineral category of the crystal. The boundaries between regions are then processed. Each interface segment is extracted in turn, labelled and stored in a third image in registration. The vertices are extracted, labelled and stored on the same image. Negative labels are used to distinguish vertices from interface segments.

Segments and vertices are identified using logical operations on the labelled region map and the original binary image, plus the basic mathematical morphology techniques of erosion and dilation [2.3.2.1]. For example, dilating two regions using a 3X3 structural element, and logically ANDing these two images with the boundary image identifies those border pixels on the interface between the regions. Vertices may be derived in a similar fashion, or by ANDing individual segment images. Dilating and eroding a region with the same 3X3 structural element, and then ANDing it with the borders identifies any internal cracks in the region.

The information on contiguity of regions, segments and vertices was extracted in this way and tabulated. The segment labels were used as key values for data access. Further details of the processing sequence have not been given, and queries regarding the implementation remain unanswered. Although a method for extracting the interface segments between regions is given, quite how the regions are known to be adjacent is unclear. Is every pair of regions tested? Apart from such queries this processing sequence does not produce all the information required for the applications described above [4.2.1]. Region boundaries are not encoded, and there is no distinction made between internal and external region neighbours. Establishing such relationships by analysing the tabulated data could prove difficult when large clusters of crystals are surrounded by one other.

The scheme implemented, to be described below [4.3.2], used a border tracking in place of dilation to achieve similar results to the GIAPP method, while simultaneously producing chain codes of region and hole boundaries, and establishing inclusion relationships.

4.3.1.2 A graph traversal-based scheme

As described above [4.2.2.3], at an early stage in this study a network structure of information describing a partially segmented petrographic image was implemented. Taking a thinned and labelled region map as input, border segments were chain coded and labelled, vertices identified and labelled, and regions related to their adjacent line segments, vertices and neighbouring regions, also preserving the distinction between internal and external neighbours. This was achieved in what was in effect a single pass over the data. The algorithm was based on a graph traversal technique.

The partially segmented crystal profiles image, such as figure 3.1, can be viewed as though it were the diagram of a graph (or multigraph). The vertices in the image represent the nodes of a graph; the boundary segments linking vertices represent the graph edges or arcs. The image is thus a diagram of its vertex adjacency graph. The implemented scheme for extracting spatial information from such an image exploited this parallel by adapting a standard graph traversal algorithm: depth first traversal [Horowitz & Sahni 1984, sect.6.2].

When appropriate data structures for describing a graph have been established, the depth first traversal algorithm can be described in only a few lines of program code through the use of a recursive formulation. The algorithm proceeds as follows. The start vertex *v* is visited. Next an unvisited vertex *w* adjacent to *v* is selected and a depth first traversal is initiated from *w*. When a vertex is reached with no unvisited vertices adjacent to it then back up to the last vertex visited and initiate a traversal from there. The traversal terminates when no unvisited vertex can be reached from any of the visited ones.

The problem in this new application of the technique, however, was that only a *picture* of the graph was given as input, and the appropriate data structures had to be created and maintained as the traversal proceeded. In addition routines for extracting the necessary structural information had to be incorporated. The version implemented used the network structure representation described above [4.2.2.3], but the algorithm could be adapted to supply data to other structures.

Before traversal the regions were labelled by a blob-colouring algorithm as described above [4.3.1]. One additional pre-processing step was used which simplified later analyses; two layers of pixels were added to the perimeter of the image. The pixels of the first layer added were 'border' pixels, i.e. zeroes, if their 4-neighbour in the image was not already a border pixel; otherwise the pixels were given the value 20, which represented the 'exterior region'. The pixels added as the outer layer were all given this exterior region label value. As a result of this process all regions in the input image had a complete, thin outer layer of border pixels; the special exterior region is the sole exception. This procedure made it relatively easy, for example, to extract useful information about which regions and boundaries had been truncated by the picture frame.

Extracting graph components

Using a graph traversal technique meant that connected components of the (vertex adjacency) graph were analysed one at a time. This led to a simple method of identifying holes in regions, since the boundaries associated with the constituent regions of a hole form a disjoint component of the VAG.

The analysis of the region map required both row-by-row scanning and tracking along border line segments. The labelled image was tracked until a border pixel was found. The border was then tracked from this point until a vertex was found and the traversal of this component of the graph was initiated - starting the traversal from a vertex simplified the algorithm slightly. If following the border in search of a vertex led back to the starting position before any vertex was encountered then it could be deduced that the scanning process had discovered the border of a hole which had no vertices on its perimeter. In this case a pseudovertex was created on the hole boundary as a reference point. The tracking algorithm recognised as vertices border pixels at the convergence of 3, 4 or only 1 border line segments. This last case corresponded to the end of an intra-region crack.

The extraction of each component of the graph used a recursive procedure, basically the depth first traversal described above. When a vertex was encountered a check was made to see if it had an incident border segment as yet untracked by the algorithm. If no such segment was found the previous vertex visited was returned to. If there were such a segment it would be explored until a vertex was reached. At this vertex the procedure would be re-invoked. This process required the establishment and maintainance of global records associated with each vertex found, detailing the progress of the traversal.

Having tracked a complete component of the vertex adjacency graph and extracted the required structural information, the algorithm continued the scan from the point where it first met the component. In this way the components were found and analysed one by one. In order that the scanning process would not pick up an already analysed part of the boundary network the boundary pixels which had been tracked along had their values changed.

Vertices were labelled according to their order, i.e. the number of boundary segments incident. Vertices where 3 or 4 segments converged were given the labels 13 or 14 respectively. A 'dead end' vertex at the end of a border segment (which must have been an intra-region crack) was given the label 11. Pseudovertices were labelled 12. The border pixels which were not vertices were given as a label the chaincode chainlet used in tracking, to facilitate any re-tracking over the boundary network. It was necessary to use a slightly different chaincode scheme from that described above: 8 was used instead of 0 to mark one possibe local direction. The value 0 was used to denote border pixels as yet unvisited by the traversal algorithm. Region pixels were labelled with values from 20 upwards, so there would be no possible confusion during tracking.

The pre-processing blob-colouring gives advance knowledge of the number of regions in the image, i.e. the *faces* of the graph. This information could be used to deduce when all components of the graph had been extracted without the need to scan every pixel in the image. Euler's formula relating the number of vertices, edges and faces in a planar graph is well known: V-E+F=2 (see, for example, section 9.3 of Bondy and Murty's text [1976]). For a graph consisting of n components the corresponding formula is V-E+F=n+1. The VAG is an n-component planar graph for which F is known. After a component has been extracted the V, E and n values found so far can be used to test the equality V-E+F = n+1. If the equality does not hold then at least one component remains to be found and so the scan may be continued. If the equality holds then every component has been found. This test requires the use of pseudovertices on hole boundaries, and depends on each graph component contributing at least one face to the graph. It would therefore only be applicable if the image were guaranteed to have no intra-region 'cracks'.

Tracking borders

Procedures were required for tracking a line in the image and recognising a vertex at any point along the way. The algorithm was designed to cope with borders of variable thinning: both 4-connected and 8-connected border segments were acceptable. Thus, for example, the border between the two regions in figure 4.4 could be handled correctly.

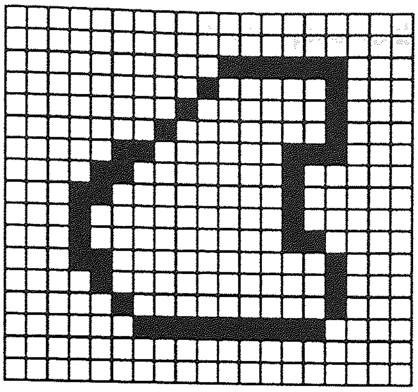


Figure 4.4 An inter-region border of variable thinning

The possibility of a border pixel having a 4-neighbour with an 8-neighbour adjacent to it can lead to errors in tracking if care is not taken. Similar problems could arise in the recognition of vertices. Possible errors of interpretation are illustrated in figure 4.5. In tracking, the problem is solved by examining 4-neighbours before 8-neighbours, and ignoring the 8-neighbours which are adjacent to any 4-neighbours found. Vertices can be correctly identified by disregarding a pixel's 8-neighbours adjacent to 4-neighbours.

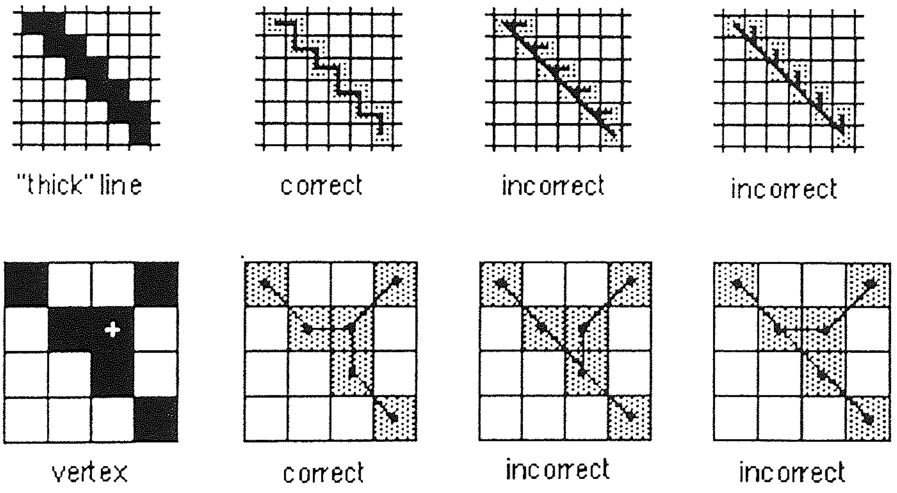


Figure 4.5 Correct line tracking and vertex recognition

During line tracking the direction in which to move on the next step depends upon the direction from which the current point was reached. If the previously visited position was a 4-neighbour, then search the opposite 4-neighbour, the other two 4–neighbours and then the two 8-neighbours adjacent to the opposite 4-neighbour. If any 4-neighbours were found, however, do not look for the adjacent 8-neighbour. If the previous point visited was an 8-neighbour then search the two 4-neighbours adjacent to the opposite 8-neighbour, then the three 8-neighbours. Again, if a 4-neighbour is found do not look for the adjacent 8-neighbour. This searching procedure was implemented efficiently using a look-up table indexed by the chain code representing the local direction from the previous pixel to the current search position.

Data extraction

The number of valid neighbours found, not counting the neighbour point just visited, reveals the nature of the point. A single valid neighbour means the point is an 'arc point', i.e. a point within a border line segment, and the tracking algorithm moves to this new point on the line and repeats its search. 0, 2 or 3 neighbours mean the point is a vertex.

When a vertex was found the value of the pixel revealed if it had already been encountered, since vertices were labelled with non-zero values when found, as described above. For new vertices a record was created tabulating the local structure: the regions adjacent, the directions of the incident border segments, which segments had been traversed, and for these segments which vertices lay at the other end. For a previously encountered vertex the record was updated with the fact that another of its incident border segments had been traversed, and what vertex it led to. The record of the vertex from which this segment had led was also updated.

A record was also created for the boundary line segment now delimited, its chain code was stored, and links established to its two terminal vertices. Region records, which were set up following the blob-colouring, were also updated. This information was stored in the network data structure, as described above. Some additional record fields were required, but since these were for internal use only they were not mentioned above.

When a new component of the graph was found a note was made of the enclosing region so that its record of internal neighbour relations could be updated. A pointer to the first vertex (or pseudovertex) in this component was used to associate it with its surrounding region. The adjacencies between the regions on the exterior of a hole and the surrounding region were identified by using the line segment information to locate the records of the vertices on the border, and examining the region-vertex incidence data held there.

The traversal scheme was efficient, in that each boundary segment was tracked only once and the tracking implementation used the look-up table method described above, and it supplied all the information required by the network database in a single pass. Its complexity and its intimate association with the network structure made it difficult to modify, however. It would be possible to partition the functions of the algorithm, separating the procedures for control of the traversal from those for data extraction. This should result in a more flexible and more easily maintained algorithm.

When the network structure was rejected in favour of the more flexible relational

SPIN database the graph traversal scheme was abandoned with it. The 'Spinster' algorithm subsequently devised for eliciting image spatial information retains some of the techniques used by the traversal scheme, but has a much simpler structure which makes it more easily understood, maintained and modified.

4.3.2 Constructing SPIN: the Spinster algorithm

The Spinster algorithm takes as input a partially segmented crystal profiles image and creates for it the SPIN Spatial Information Database which describes the important structural features and their interrelationships.

The input is assumed to have been pre-processed in a manner similar to that demanded by the graph traversal scheme. This requirement means that border segments must be thinned and regions labelled with the values 21 upwards. Two pixel layers should also be added to the perimeter of the image, so completing the border of regions truncated at the edge of the image and supplying the 'exterior region' to simplify later processing.

The Spinster algorithm overwrites the image in a different manner from the traversal method, and the value 10 is used to mark the exterior region with no risk of confusion. The chain coding scheme illustrated in figure 4.2 is used. Like the traversal method, Spinster is designed to cope with both 4-connected and 8-connected border segments. However, the requirement that it should handle intra-region cracks was dropped. The algorithm which was designed and implemented does not function correctly if these features are present and they must therefore be removed. The question of how the algorithm can be extended to cope with intra-region cracks is considered below [4.3.2.5].

4.3.2.1 Outer and inner region borders

In section 4.3.1.1 the use of dilation and simple logical operations for the classification of border pixels was explained. The techniques described there did not, however, discriminate between internal or external borders. Further, the classification made no attempt to address the problem of contour encoding and the additional application of a contour following algorithm would be required to provide the sort of information a useful shape analysis demands [4.2.1].

In this section it will be demonstrated that a contour following algorithm can be adapted so as to provide the same sort of information generated by the dilation technique used by GIAPP. This original technique also permits the classification of the borders as internal or external, which is not possible by simple dilation-based methods alone. Since this is achieved while chain-encoding the borders the technique is also seen to have efficiency advantages over the two-stage process mooted in the previous paragraph. This technique is the foundation of the Spinster algorithm.

Every region in a crystal profiles image (except the exterior region) has one closed outer contour. This is true even for regions apparently cut off by the frame, because of the pre-processing described in the previous section. A region will have an inner border or 'hole' border for every inclusion or hole it contains. The hole border itself must also be the outer border of a region, or perhaps be made up of parts of the outer borders of more than one region if the hole has two or more component regions adjacent to the surrounding region.

The augmented tracking technique used in Spinster may now be described. Each region in the image is visited once, and its outer contour is chain-encoded. At the same time, the pixels on the region border have their values incremented by one. When every region has been visited the border pixels, which initially all had value zero, may now have the values 1, 2, 3 or 4. There can be no pixels of value zero remaining.

Pixels of value 1 must lie on the outer border of one region only, and hence lie on a hole border of another region. Pixels of value 2, 3 and 4 lie on the outer border of, respectively, two, three and four regions. The values 3 and 4 therefore indicate vertices in the image. (Notice also that at this stage the value 2 could indicate a vertex on a hole border). The important result is, however, that there is a pixel of value 1 in the image if and only if that pixel lies on a hole border in the image.

A second pass may now be made over the image, seeking out pixels of value 1. When such a pixel is found, the hole contour is chain-encoded, and the border pixels incremented as before. The next hole may then be sought, via its tell-tale pixel values. At the end of this second pass all the border pixels will have the values 2, 3 or 4. The values 3 and 4 will mark vertices and, conversely, all vertex pixels will have such values; the value 2 marks all other border pixels.

4.3.2.2 The boundary tracking algorithm

The boundary tracking algorithm used may be viewed as an extension of the 'crack-following' algorithm described by Rosenfeld [Rosenfeld & Kak 1982, p220], although it was developed independently. The algorithm finds the border pixels which are 4- or 8-neighbours of a labelled, 4-connected region. The procedure for clockwise encoding of the contour will be given; the procedure for the counter-clockwise encoding is derived similarly.

The first step is to find a suitable start position: namely, any region pixel which has a 4-adjacent border pixel. A simple row-by-row scan through the image is performed in order to find a pixel bearing the label of the region. If the first pixel of the region is encountered at (xr,yr), then it follows from the input data assumptions that the 4-neighbouring pixel at (xr-1,yr) must be a border pixel. This border position may be called (xb,yb). These two pixels are suitable starting points: figure 4.6 illustrates the situation.

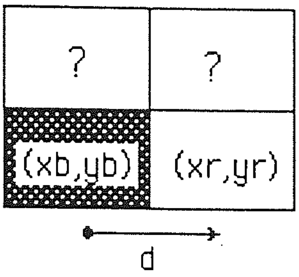


Figure 4.6 Boundary tracking start conditions

The direction of the point (xr,yr) with respect to (xb,yb) is denoted by d, and this may be represented by a chain code element. Here at the beginning d=0, but in general d∈ {0,2,4,6}, since the two pixels are 4-neighbours. One step in the tracking may now be made. Only two pixels need be examined at each step.

If one imagines oneself standing on the crack between the border and region pixels just identified, facing in the direction such that the region pixel is on the right hand side, then the two pixels which lie immediately ahead are examined. Only 4 possible configurations can arise (at all orientations): these are illustrated as cases 1 to 4 in figure 4.7.

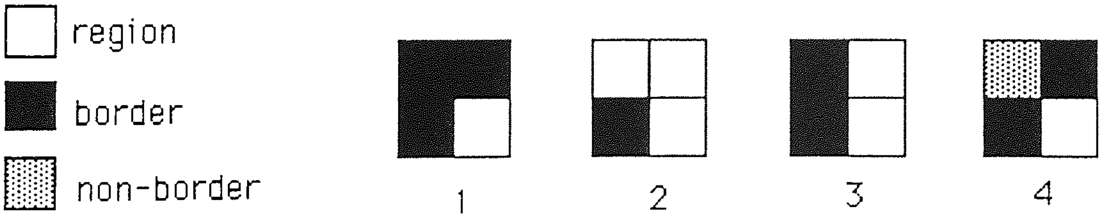


Figure 4.7 The four possible situations during tracking

Case 1 corresponds to a turn to the right. In this situation two code chainlets are generated: d-2 (modulo 8) and d. The region pixel coordinates (xr,yr) are left unchanged, but the border pixel coordinates are altered, moving one step in the direction d-1 (directions may henceforth be assumed to be taken modulo 8). The direction d is then itself updated becoming d+2.

Case 2 corresponds to a turn to the left. In this situation no code chainlets are generated, the border pixel coordinates are left unaltered, but the region pixel coordinates are updated, moving one step in the direction d-3. The direction d is updated to become d-2.

Case 3 corresponds to a single step forward without turns. One code chainlet is generated: d-2, and both the border and region pixel coordinates are updated to take them one step in the direction d-2. Direction d is left unaltered.

Case 4 corresponds to a turn to the right, but the border is not 4-connected (c.f. case 1). One code chainlet is generated: d-1. The region pixel coordinates are left unaltered, the border pixel coordinates and the direction d are both updated as in case 1.

After handling the appropriate case as described, the (xb,yb), (xr,yr) and d values have been updated, and the process continues. The algorithm terminates when the updated values are the same as the start values. The chaincode generated will describe the (external) border of the region in a clockwise direction. It is easy to see how this algorithm could be adjusted to track the border in a counter-clockwise direction. It is a trivial matter to alter the labels of the border pixels encountered during the tracking, as is required by the Spinster algorithm.

4.3.2.3 Spinster: implementation details

The Spinster algorithm extracts key structural and spatial information about the input image in a relatively efficient manner, through the use of the adapted border tracking technique described above. The complete strategy will now be explained. There are four principal steps.

The first stage is the examination of each region's outer contour. A row by row scan is used to find a (non-exterior) region. A note is made of the label of the region found. The region's outer contour is then chain encoded (clockwise), and the border pixels incremented by 1. Database region records and outer contour records are created at this time, and the information recorded there [4.2.3]. The scanning then continues from where it left off until a new region is found, and the process repeats until all regions have been visited. At the end of this processing stage all regions' outer contours have been encoded, and the database region records and outer contour records will have been completed.

In the second stage each hole in the image is found and examined in turn. A similar search procedure is used, but now the pixel value 1 is sought, since this indicates a hole contour [4.3.2.1]. The hole contour is then chain encoded (anticlockwise), and the pixels incremented by 1. Spinster checks the database to see how many holes have already been found for this region; a database hole contour record is created for each hole found, and indexed by the label of the containing region and the hole number [4.2.3].

In the third processing stage each outer contour is revisited in turn, using the information now stored and available in the database. The pixels of the contour are examined, and a note is made of all the vertices on the contour (i.e. pixels of value 3 or 4). Database records are created for each new vertex, giving it a unique label, storing its location and noting the region on which it lies. The database must be checked to see if this vertex has already been found when tracing another contour, which would mean that a vertex record already exists. If the vertex record is already present it is updated with the new incidence information. It is at this stage that the vertices are related to the contours on which they lie by creating outer contour / vertex records [4.2.3] which store the vertex positions in terms of chainlet numbers in the contour chaincode.

Now Spinster extracts region adjacency information - finding the *external* neighbours of the region whose contour is currently being visited. This is achieved by inspecting the local neighbourhood of each vertex on the outer contour. An external neighbours record is created for the region, recording the adjacent regions, and the pairs of vertices which define each interface are also recorded [4.2.3]. When outer contours have no vertices on them it may be deduced that the region is wholly surrounded by one single region. In this case no adjacency is recorded in the database, but the situation is fully dealt with in the subsequent stage.

In the fourth and final processing stage all the hole contours are re-visited, using the information stored and available in the database. The pixels of the contour are examined, and a note is made of all the vertices on the contour. All these vertices must have already been encountered on the outer contour of some region, and so a vertex record must already exist in the database. The appropriate record is searched for, using the location information, and when found the record is updated with the new incidence information. The vertices are again related to the contour by creating a hole contour / vertex record [4.2.3] which stores the vertex position in terms of chainlet numbers in the contour chaincode.

Since all boundary pixels (excluding vertices) lie between two regions they will have been encoded twice, in opposing directions, as part of two outer contours or one outer contour and one hole contour. The vertex information recorded in the database is used as a way of relating the two representations of a boundary segment or complete contour: the vertices pin the two chaincodes together. When one region is wholly surrounded by another and no vertices are present on the interface another method of relating the two representations of the interface must be used. One direct solution would be to simply use the same chain code (and origin point) for both representations of the contour. However, it seems preferable to maintain consistency over the image by keeping the two code representations in different directions. The solution used is therefore to retain two different chaincode representations of the contour, one clockwise and one anticlockwise, but to provide the link between them by adjusting one so that they have a common origin point.

Finally, Spinster extracts region adjacency information - finding *internal* neighbours for the region which contains the hole, by identifying the regions in the hole which lie adjacent to the hole boundary which is currently being visited. This is achieved by inspecting the neighbourhood of each vertex on the hole boundary. An internal neighbours record is created for each region which has at least one hole and, as was described above [4.2.3], records for each hole are stored as sub-tables. Here are recorded the adjacent regions, and the pairs of vertices which define each interface. When there are no vertices on the interface it is at this point that the adjacencies are recorded: an examination of the border neighbourhood identifies the regions involved, and then both the internal and the external neighbours records concerned are found and updated appropriately. The fact that only one neighbour has been found for a region allows simple inclusion to be deduced.

The database of information described above [4.2.3] is now complete.

4.3.2.4 SPIN database example

Figure 4.8(a) shows a test image for which a spatial information database has been constructed using the Spinster algorithm. Figure 4.8(b) shows the region labels given by the pre-processing blob-colouring algorithm. Also indicated in bold type are the vertex labels assigned by the Spinster algorithm. Table 4.1 lists the eight tables of the SPIN database representing this partially segmented image. The ellipsis (...) indicates editing of the chain codes.

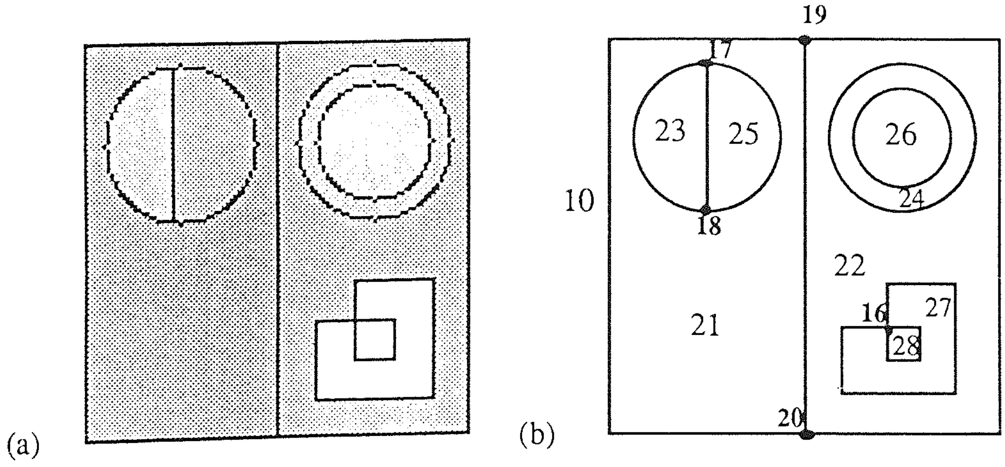


Figure 4.8 A test image and the labelling used in the SPIN database

Region table

```
[[28 []] [27 []] [26 []] [25 []] [24 []] [23 []] [22 []] [21 []] [10 []]]
```

Outer contour table

```
[[28 73 74 40 [6 0 0 ......6 6 6]]
 [27 73 64 160 [6 0 0 ......6 6 6]]
 [26 78 13 96 [1 0 0 ......0 0 7]]
 [25 21 10 98 [7 0 0 ......0 7 0]]
 [24 78 8 128 [1 0 0 ......0 0 7]]
 [23 27 9 106 [7 1 0 ......6 6 0]]
 [22 53 4 296 [6 0 0 ......6 6 6]]
 [21 3 4 298 [6 0 0 ......6 6 6]]]
```

hole contour table

```
[[22 2 73 63 120 [2 2 2 ......4 4 4]]
 [24 1 78 13 96 [3 4 4 ......4 4 5]]
 [22 1 78 8 128 [3 4 4 ......4 4 5]]
 [21 1 28 8 128 [3 4 4 ......4 4 5]]
 [10 1 3 3 396 [2 2 2 ......4 4 4]]]
```

vertex table

```
[[20 53 102 3 [10 21 22]]
 [19 53 3 3 [10 21 22]]
 [18 26 47 3 [21 23 25]]
 [17 26 9 3 [21 23 25]]
 [16 73 73 4 [22 27 27 28]]]
```

outer contour / vertex table

```
[[28 1 [[16 1]]]
 [27 2 [[16 151] [16 111]]]
 [26 0 []]
```

...continued over

```
[25 2 [[18 43] [17 5]]]

[24 0 []]

[23 2 [[17 105] [18 67]]]

[22 2 [[20 198] [19 1]]]

[21 2 [[20 150] [19 51]]]]
```

hole contour / vertex table

```
[[22 2 1 [[16 10]]]

[24 1 0 []]

[22 1 0 []]

[21 1 2 [[18 62] [17 2]]]

[10 1 2 [[19 346] [20 149]]]]
```

external neighbours table

```
[[21 2 [10 22] [[20 19] [19 20]]]

[22 2 [21 10] [[20 19] [19 20]]]

[23 2 [21 25] [[17 18] [18 17]]]

[24 1 [22] []]

[25 2 [21 23] [[18 17] [17 18]]]

[26 1 [24] []]

[27 2 [22 28] [[16 16] [16 16]]]

[28 1 [27] [[16 16]]]]
```

internal neighbours table

```
[[24 1 [[1 1 [26] [[]]]]]

[22 2 [[1 1 [24] [[]]] [2 1 [27] [[16 16]]]]]

[21 1 [[1 2 [23 25] [[18 17] [17 18]]]]]

[10 1 [[1 2 [21 22] [[19 20] [20 19]]]]]]
```

Table 4.1 A SPIN database for figure 4.8

4.3.2.5 Future work: coping with cracks

The applicability of the Spinster algorithm and the SPIN database could be broadened and the information extraction made more robust by the inclusion of routines to handle the presence of 'cracks' in the image, i.e. border pixels in the thinned, segmented region map having the same region on both sides. Cracks may be considered in two categories: border cracks which extend to the outer edge of a region, and internal cracks which do not. Internal cracks may extend to the border of an internal hole; border cracks may extend from a region's outer border to an internal hole border. Both types of cracks may branch off in tree-like structures, but they enclose no regions. Figure 4.9(a) illustrates some border cracks, shown darker; 4.9(b) illustrates some internal cracks.

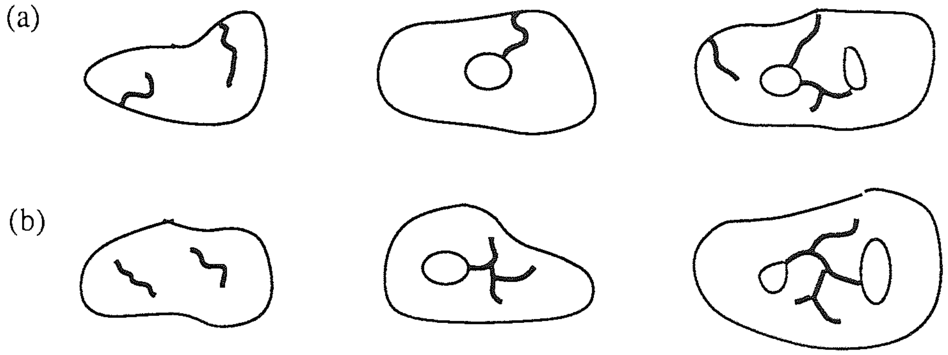


Figure 4.9 Border and internal cracks

These features could be handled by an extended version of the border tracking scheme currently used in the Spinster algorithm. The key modification is the use of a second array in registration with the image array.

In the first stage of the Spinster algorithm each region is sought and its outer contour is chain encoded, incrementing the values of the border pixels passed over. The alternative strategy is to track and encode the border as before, but to increment the pixel values of the corresponding positions in the parallel array, not the image array. When the region border has been fully tracked and encoded the chain code produced may be used to repeat the circuit of the region profile in the parallel array. In this image the pixel value 2 indicates a crack pixel.

Figure 4.10(a) gives a simple example of a region boundary segment with one branching border crack extending into the region, with the post-tracking pixel values indicated.

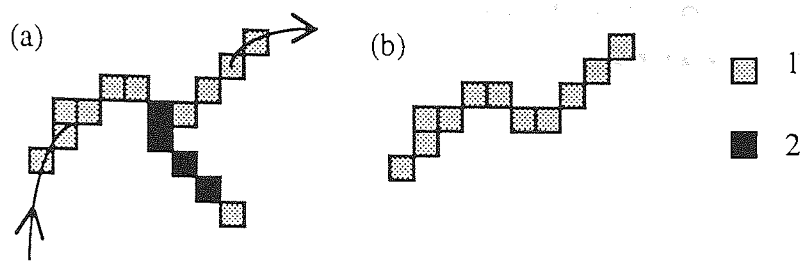


Figure 4.10 A border crack on a boundary segment, and its removal

At this point the crack could be absorbed into the surrounding region by overwriting the crack pixels with the region label, and the perimeter pixels and chain code could be adjusted accordingly. This would result in the parallel image having the form shown in figure 4.10(b). Using the new chain code the corresponding pixels in the original image array may now be incremented, and the Spinster algorithm continued as before.

An alternative strategy is to record the presence of cracks in the SPIN database. This would require new data structures. It is anticipated that new primitives such as crack segments and crack vertices would be needed, together with structures recording the relations between a crack, the surrounding region and any contiguous holes. The problem of representation has not been considered further.

The analysis of the parallel image produced by tracking will be complicated by the possible presence of complex features such as those illustrated in figure 4.9(a). Checks will be required to find whether cracks lead to internal holes in the region. Figure 4.11(a) shows how the perimeters of holes would be tracked, if linked by fortuitous cracks. The corresponding pixel values found when following the resultant chain code are sketched in figure 4.11(b). Correspondences between key points in the image and the coded sequence are indicated.

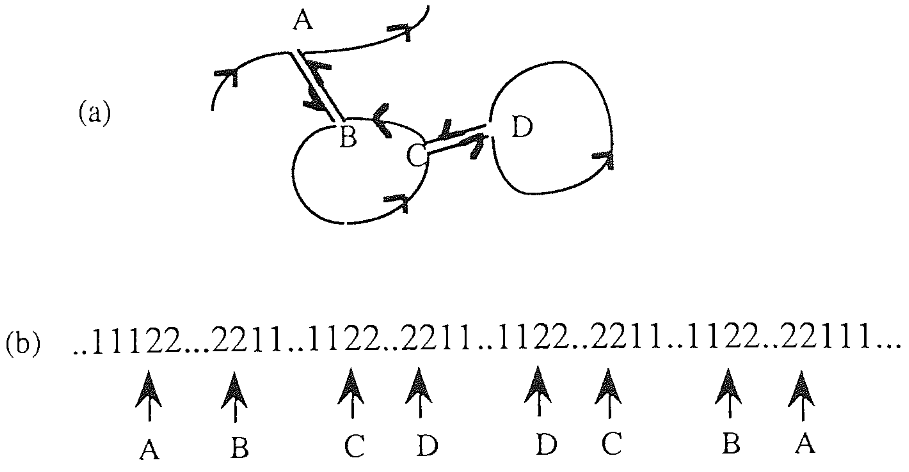


Figure 4.11 Tracking border cracks around holes

The nested arrangement of the coded sequences can be seen. A check on the image coordinates corresponding to the ends of sequences of 2s found when following the chain coded path should lead to the recovery of the structure.

Internal cracks, such as those illustrated in figure 4.9(b), would not interfere with the further operation of the Spinster algorithm and so their presence could be ignored unless a description was specifically required. They could be identified by searching the image array for the pixel value zero, following the first pass to encode and increment region outer contours.

The tracking and incrementing process using a parallel image array should again extract each crack and all its possible branches. The value 1 in the image would indicate the end of a crack; the values 3 or 4 would indicate crack junctions; the value 2 would indicate the pixels of the rest of the crack network. Thus the information required for unravelling the chain code and building a structural description could be extracted.

4.4 Conclusions

This chapter has addressed the problems of extracting and representing key structural and topological information to describe a partially segmented image. The structure of SPIN, a relational database for spatial information, has been described in detail and set in context through a discussion of related work. A scheme for extracting the required information has been described. This combines chain code generation and the elicitation of adjacency and inclusion relations, through replacing morphological dilation with border tracking. A useful extension of this approach has been discussed.

The spatial information extracted and stored is required for image analysis routines which will be described in subsequent chapters. However, as the discussion of the previous chapter has indicated, such information is fundamental to many image analysis tasks. The algorithms and data structures developed here may therefore be relevant to other image analysis applications, such as remote sensing or histology.

CHAPTER 5

Multiscale Contour Analysis

5.1 Introduction

In this chapter techniques for identifying faces or possible faces on a single crystal profile or 'contour' are investigated. The requirements for a suitable contour analysis scheme are presented. The nature of the problems involved in crystal profile analysis are such that no published scheme for contour analysis has been found which is completely appropriate or adequate, and a new method using a multiscale approach is proposed. An implementation of these ideas is described in detail. The new algorithm devised produces a multiscale analysis of the linear and near-linear structure on a contour in the form of a hierarchical structure which is termed the 'face tree'.

The face tree provides a useful representation of the contour structure which in some situations may be an adequate analysis. It is also a suitable basis for further analyses using non-local contour properties such as symmetry or collinearity. The following chapter describes new algorithms for crystal symmetry analysis which operate upon the face tree representation. The algorithms presented in chapters 5 and 6, although not developed to a final, polished form, together appear to provide a solution to the principal problems of crystal profile analysis, and constitute a novel contribution to contour analysis in general.

5.1.1 Problems in crystal profile analysis

As has been established previously [2.2.5], crystal faces are realised as straight line segments in the contours. However, crystal development may not be perfect; steps in the image processing to extract crystal contours may also introduce small errors. As a result of either or both of these factors the actual faces will almost certainly not be perfect line segments. Some departure from the ideal state must therefore be accepted.

In considering approximations to curves it is useful to introduce the idea of 'resolution', which in this context refers to the allowed range of transverse deviations of the original points from the approximating curve. In a high resolution approximation very little deviation is permitted; in low resolution approximations greater deviations may be accepted. Thus perfectly acceptable crystal faces on a crystal profile may be quite irregular at high resolution.

In analysing crystal profile images the appropriate resolutions for identifying faces cannot be known in advance. Further, the optimal resolution may easily vary from point to point on the contour, reflecting changes in the character of the crystal profile. These two factors mean that the 'obvious' approach to contour analysis - using a conventional, fixed-resolution algorithm for polygonal approximation and treating the polygon sides produced as crystal faces - will in general be unsuccessful.

There is an additional important factor which must be considered: the influence of the figure as a whole on local interpretations. It has been argued above that, in petrography as elsewhere, the perception of the whole figure is important, and influences the interpretation of the elements in the figure. Thus any attempt to find crystal faces on a contour which relies only on local measures such as segment straightness is likely to be inadequate in some situations. It would appear that that this factor has not been considered in published algorithms for contour shape analysis; it has not been considered in the context of crystal profile analysis.

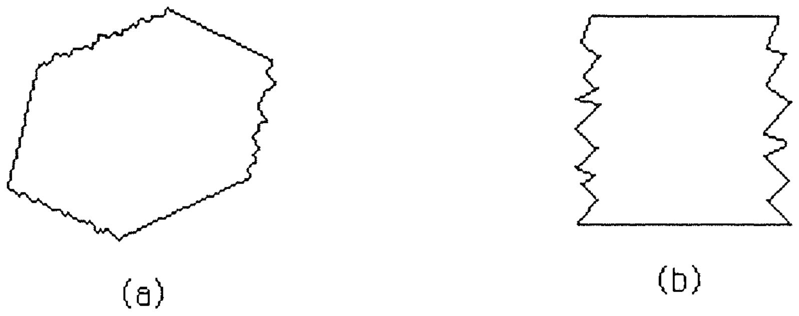


Figure 5.1 Contours showing structure at more than one resolution

Figure 5.1 illustrates some of these points. Figure 5.1(a) is perceived as hexagonal although the sides are not all perfectly straight and show different degrees of irregularity; figure 5.1(b) is basically rectangular, but the vertical sides show additional linear structure at higher resolution.

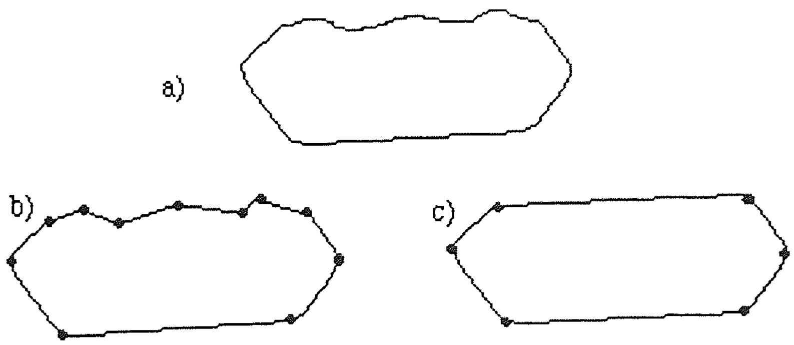


Figure 5.2 Alternative polygonal approximations of a contour

Figures 5.2(b) and (c) give two possible polygonal approximations for the contour of figure 5.2(a). On the basis of local measures of contour straightness it would be hard to decide which is the more acceptable approximation. However, when the symmetries of the whole figures are taken into account the alternative of 5.2(c) is seen to be preferable.

Thus finding faces on a crystal profile is a perceptual task which is more complex than might at first be assumed. It would appear that significant linear structure must be sought over a range of resolutions. A measure of the straightness of each side found is also required: for use in the symmetry analysis and also to quantify the irregularity or 'quality of development' of faces, which a complete petrographic description demands. Some mechanism must also be found which can support the additional use of symmetry information in the analysis.

The principle proposed here is that where there is reasonable ambiguity regarding the linear structure of the contour the various possibilities should all be passed on to be resolved by subsequent analyses which use symmetry information.

5.1.2 Chapter outline

Section 5.2 reviews published work on creating polygonal approximations to contours, and methods for curve partitioning which use analyses at different resolutions or scales.

Section 5.3 describes a new algorithm for multiscale analysis of contours which aims to solve the problems outlined above [5.1.1]. The algorithm produces a new intermediate representation of the contour structure in the form of two two-dimensional arrays which carry 'corner' and 'side' information. Sections 5.3.1 and 5.3.2 show how the intermediate representation is created and describe its principal features. Section 5.3.3 shows how the significantly linear portions of the contour (i.e. possible faces) can be detected via the intermediate representation. Section 5.3.4 shows how the collection of suggested faces extracted from the intermediate representation can be reduced and refined to produce a hierarchical description of the possible faces. Section 5.3.5 shows how the intermediate representation can also be used to create a 'conventional' fixed-error polygonal approximation to the contour. Section 5.3.6 reviews the development of the new contour analysis algorithm and describes the current form of the program implemented. Section 5.3.7 gives implementation details for the principal stages of the algorithm.

Section 5.4 is concerned with examining the performance of the new algorithm. Section 5.4.1 considers the problems arising in measuring the effectiveness of such an algorithm, and presents the scheme which was devised for this purpose. Sections 5.4.2 and 5.4.3 present test examples and analyse the results, including comparisons with the results produced by another algorithm and the results produced by human experts. Section 5.4.4 examines the efficiency of the algorithm.

Section 5.5 reviews the chapter and gives recommendations for further work.

The algorithm presented in this chapter does not in itself provide a compete solution to the problems of crystal profile analysis described above. The analysis is concerned only with measurement of local straightness on the contour, and it has been shown that non-local symmetry information should be included for a comprehensive analysis. However, the algorithm produces a description of the possible faces in the form of a hierarchical data structure, termed the face tree, which is a suitable basis for symmetry analysis. The following chapter presents new algorithms for the symmetry analysis of face trees. Thus chapters 5 and 6 together offer a solution to the central problems of crystal profile analysis.

5.2 Contour analysis: a review

Two different emphases in contour analysis can be discerned: straightforward data compression and simplification, and what may be described as pattern recognition, where the aim is the recognition of features such as 'sides' or 'corners'. The following review considers published schemes in two corresponding groups.

5.2.1 Fixed resolution polygonal approximation of contours

Replacing a digital image by an approximating polygon brings the benefits of efficiency of representation and opportunities for more sophisticated analysis and manipulation of the data. Consequently many algorithms have been developed for finding polygonal approximations to a closed contour. Some representative examples will be briefly considered here; the papers of Kurozumi and Davis [1982], Pavlidis [1982] and Dunham [1987], for example, provide more detailed overviews of polygonal approximation algorithms.

The process of approximating a closed contour by a polygon is special case of the problem of finding a piecewise linear approximation to a curve. Some algorithms constrain the vertices of the approximating polygon to lie on the original curve, which simplifies implementation but may have no other justification. It may be acceptable simply to partition the contour into sections each of which is an approximation to a straight line segment, with no requirement on the continuity of the overall approximation which results, in which case the approximation should be termed segmental rather than polygonal.

The overall fit (or error) of the approximation is derived from the fit of each segment of the curve to its approximating line. Various measures of fit are possible.

If the distance from a point i on the curve to the straight line approximating it is denoted by e(i), then two possibilities are to seek to minimise the sum of the magnitudes of these distances for all points: $\Sigma|e(i)|$, or the sum of the squares of these distances: $\Sigma e(i).e(i)$. These are termed respectively the l1 and l2 error norms. Kurozumi and Davis presented the 'minimax' method for polygonal approximation which used as its error measure the maximum $|e(i)|$ value. This choice of error norm, sometimes called the the $l\infty$ norm, has the advantage over the 'least-squares' (l2) method in that it does not miss details of the data such as a spike - i.e. a change of short duration. In the minimax algorithm the vertices are not constrained to lie on the original curve.

Various constraints may be placed on the approximation, in addition to the error being below a given threshold. One may seek, for example, the approximating polygon with the least number of sides (e.g. Dunham's [1987] method), or the polygon of minimum perimeter [Sklansky 1970]. The number of sides may be unimportant or specified in advance (a possibility also discussed by Dunham). The implementation may consider all the data prior to selecting the points which define the approximating polygon, or may be a 'scan-along' algorithm where the contour is followed and a decision made at each point whether to begin a new segment of the approximation (e.g. Tomek's [1974] method).

Ramer [1972] gives a simple recursive procedure for producing polygonal approximations to two dimensional curves. This is a splitting algorithm. Starting with an initial, possibly arbitrary polygonal approximation, each segment is examined to see how well it fits the curve. The error of the fit is taken to be the maximum $|e(i)|$ value for the segment. If this value exceeds some fixed threshold then the point on the curve furthest from the approximating line segment is chosen as a new vertex, thus replacing the segment with two new approximating line segments. The fit of each new segment is then checked. The process stops when each straight line segment is close enough to the curve.

Pavlidis and Horowitz [1974] described an extension of this sort of method: the split and merge algorithm. The first stage of the algorithm is a splitting phase which, unlike Ramer's, used the l2 error norm. The additional steps are a merging of adjacent segments, if this merge produces a segment satisfying the error criterion, and an optional endpoint adjustment step which varies the position of the vertices slightly to see if a better fit could be achieved. Many variants of this algorithm are possible.

All these methods use the same error measures and thresholds at all positions on the curve. The use of fixed thresholds could lead to crude approximations of some small figures and unnecessarily refined approximations of larger ones. As such, these methods must be inadequate for the task of recognising structure at different resolutions. An algorithm such as the split and merge scheme could be applied repeatedly with successively greater error thresholds, but the questions would remain of how to establish appropriate error limits and how to combine the various results obtained using these values. (Bengtson's work [5.2.2.5] has addressed these problems)

5.2.2 Multiple resolution contour analysis

An examination of published approaches to multiscale descriptions will be prefaced by a consideration of the concept of scale-space, since this provides a representation which makes explicit the stability of structures appearing in (one-dimensional) curves when viewed at different scales of resolution. Scale-space diagrams allow hierarchical decomposition of arbitrary curves in a perceptually satisfying manner. The relevance of this work to the problems involved in crystal profile analysis should therefore be clear.

5.2.2.1 Scale-space

Witkin [1983] introduced scale-space and scale-space filtering as a mechanism for signal description that manages the ambiguity of scale in an organised and relatively natural manner. The 'ambiguity of scale' arises because every setting of a scale parameter (such as choosing the mask size when smoothing by convolution) yields a different description: new extrema, or other features of interest, may appear, existing ones may move or disappear, and there remains the problem of deciding which description is 'correct'. The features of particular interest to Witkin are the inflections of a signal, marked by zero crossings of the second derivative of the signal.

The scale-space filtering solution proposed by Witkin begins by 'expanding' the signal, convolving it with a series of Gaussian filters representing a range of σ–parameters, and so sweeping out a surface called the scale-space (x-σ space) image. In this representation it is possible to track zero-crossings as they move continuously with scale changes, and to identify the singular points at which new zero-crossings appear. Using these two pieces of information it is possible to reduce the scale-space image into a tree structure. If an interval is drawn in scale space at the σ–value where a singularity appears, extending in the x dimension between the two x–values to which this singularity can be tracked through scale-space, then these intervals correspond to the nodes in what is called an 'interval tree'.

The interval tree can be represented as a rectangular tesselation of scale-space, where each rectangle is a node, indicating an interval on the signal, and the scale interval over which the signal interval exists. The interval tree provides a concise description of the signal over all scales of observation. Some additional pruning of the interval tree is possible, enhancing stable features, which survive over a broad range of scales. This then produces a representation more closely corresponding to the human perception of the signal.

Figure 5.3, due to Witkin [1983], shows a signal and the associated interval tree tessellation of scale-space.

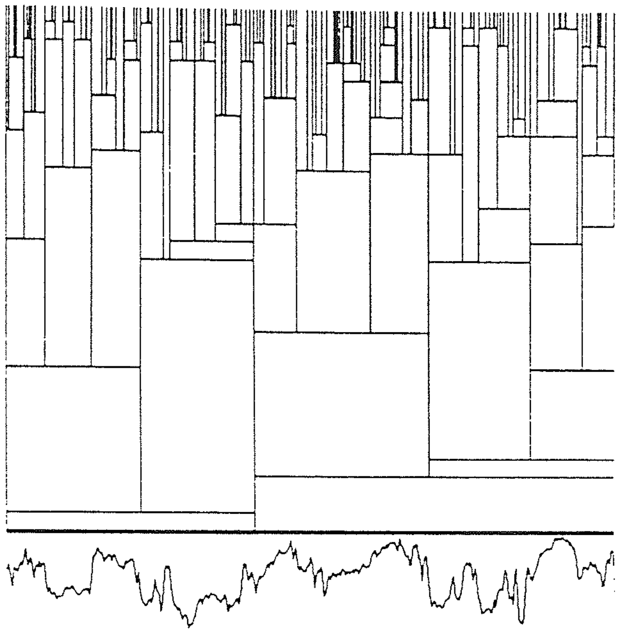


Figure 5.3 A signal and its interval tree represented as a tessellation of scale-space

Mokhtarian and Mackworth [1986] extended the ideas and techniques of Witkin to provide scale-based descriptions of planar curves and two-dimensional shapes. Unfortunately, as the authors concede, the scale-space image is not a very suitable representation for curves which consist of long straight segments, or curves which are convex to begin with - both of which eventualities are likely to occur with crystal profiles.

Bengtson *et al.* [1986] have remarked that in scale-space filtering the smoothing is indiscriminate, applied independently of the structure of the original data. As a result important features appear blurred at coarser levels, and at finer levels the structure is hard to find since it is represented with fine details and noise. The precise descriptions of a corner or linear part of the contour may not exist at any scale. The problem remains of removing unnecessary detail without destroying localisation accuracy. Bengtson's work on contour analysis will be discussed below [5.2.2.5]

5.2.2.2 Davis' angle and side detection

Davis' [1977a] paper was one of the first to address the problem of shape description at various resolutions. Davis developed methods of detecting angles and sides using approximations to the curve that have varying degrees of coarseness, and constructed a hierarchy of angles and sides aimed at describing the curve at any desired level of detail and achieving natural polygonal approximations. Two schemes were developed, dealing separately with feature points (angles or corners) and sides (relatively straight segments of the contour).

The scheme for detecting angles uses the k-curvature measure first described by Johnston and Rosenfeld [1973]. This is the cosine of the angle between the two vectors drawn from a point to its neighbours k chain-code chainlets distant on either side. Defining the domain of an angle found at resolution k to be the section of contour points within a distance k points, the angle detection scheme is based on the observation that angles are local maxima of the curvature measure, and they have uncontested reign over their domains.

The angle detector algorithm, 'ADS', proceeds by first computing the k-curvatures at every point on the curve, for a range of k values. This creates a two-dimensional array of curvature information, with each row representing one k value. The rows of this array are then treated one at a time. First all local maxima among the curvature values are found for the smallest k value, and marked as verified local maxima (VLMs). Then the second row is considered. All local maxima are found for this resolution k, and a check made to see if they are maximal amongst the curvature values for curve positions within a distance k of the point. If so, such maxima are labelled as hypothesised local maxima (HLMs). Each HLM is tested using the higher resolution results: has a smaller k value produced a curvature of significant magnitude for positions slightly more than a distance k from the point? If no such point is found the HLM is marked as a VLM. If a HLM fails the test then it is deleted and since it may have suppressed a true local maximum all points within a distance k are rechecked. This verification of HLMs has the effect of delaying to the lowest possible resolution the recognition of 'major' corners. This procedure is repeated for the next row of the curvature matrix, and terminated after the final k value is reached.

The next stage is to interpret this information: the angles are used to construct a hierarchical graph description of the sides of the contour. The graphs are tree-like and would be true trees were the contour not a closed curve. Nodes at higher levels in the graphs represent the faces defined by the more significant angles - such as the corners of the 'rectangle' in figure 5.1(b). Lower levels represent structure within these sides. Figure 5.4(a) reproduces a figure from Davis' paper, showing a roughly triangular shape; figure 5.4(b) shows the corresponding graph structure.

The graph-building procedure aims to build a hierarchy of 'plans', defined as sets of angles and their domains. First the top level plan is constructed, providing a coarse description of the contour. The algorithm proceeds recursively for each of the sides of this plan, using only angles that fall on the given side and that were not already used, until all angles found at the angle detection stage are included in some plan. This results in the required hierarchical structure.

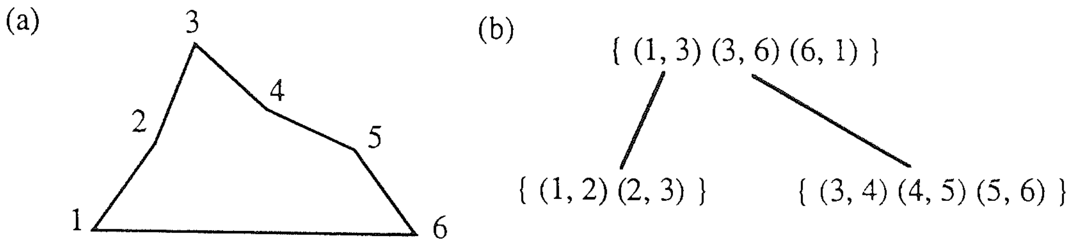


Figure 5.4 Example of a hierarchical curve structure and its graph

Davis also presented a complementary approach to shape description based on directly determining the sides of a contour. He observed that 'side' can be considered a dual notion to 'angle' because an angle is determined by a pair of sides, and a side is determined by a pair of angles. A good side is viewed as a local minimum with respect to a goodness of fit measure (of lines to arcs of given size on the contour), and consistent with the local minimum best-fitting lines to smaller arcs which have significant overlap. Davis' side detector algorithm, 'SDS', proceeds by first determining the best fit least-squares line at every point on the contour, and for a range of lengths of arcs centred at the point. In practice Davis used a series of lengths successively doubling in magnitude. For each length of arc used, the set of lines which fitted the best is found. Then a top-down filtering criterion is applied: starting with the longest hypothesised sides, a best fitting line is retained only if there exists a best fitting line half its size which overlaps it by at least half its length and which has approximately the same direction.

This procedure provides a set of best fitting lines which are used in a covering algorithm similar to that used for angles. Angles may be extrapolated from the covering and correspond to intersections of adjacent lines. This is the dual of the angle detector process, where sides were wholly defined by their vertices.

Davis' algorithms can provide descriptions of a contour reflecting significant structure over a range of resolutions. They do not, however, satisfy all the requirements of an algorithm for crystal profile analysis. It is not clear whether the appropriate approximations can be extracted from the overlapping covering sets produced by SDS. The use of ADS alone is unlikely to reveal variations in small-scale structure along a relatively straight side. The selection of angles in ADS gave no direct indication of the straightness of the sides resulting, only that they were relatively free of comparable angles. An assumption of symmetric domains for angles was also used in ADS, which might lead to incomplete description of some sides.

It may be emphasised that the two dual approaches presented by Davis were not used cooperatively. In contrast, the new algorithm for contour analysis which will be described below [5.3], uses both angle and side information.

5.2.2.3 Curve partitioning

As part of a general discussion of perceptual organisation and curve description Fischler and Bolles [1986] presented a scheme for curve partitioning based on detecting local 'discontinuity', by which is meant points which show a discontinuous or rapid change of slope. Their algorithm seeks to label each point on a curve as belonging to one of three categories: a point in a smooth interval, a point in a noisy interval, or a critical point - a 'corner'. To make this choice, the algorithm analyses the deviations of the curve from a chord that is advanced point by point along the curve. This is done for a range of chord lengths, which effectively analyses the curve at different resolutions.

If the curve stays close to the chord then points in the interval spanned by the chord will be labelled as belonging to a smooth section. If the curve makes a single excursion away from the chord, beyond some threshold, the critical point is placed at the point in the interval furthest from the chord, where the distance is summed over all chords having the point on their arcs, with the point furthest from the chord. If the curve makes two or more excursions beyond some threshold then points in the interval should be labelled as noise points. The thresholds used are some unexplained functions of the expected noise along the curve and chord length.

The classification of points is likely to vary with resolution, and 'noisy' intervals at low resolution (large chord length) will have many critical points at higher resolution (small chord length). The algorithm works from coarse to fine resolution, i.e. with decreasing chord lengths. For each chord length critical points are found and ordered by size. This list is truncated when smaller critical points begin to be found too close to larger ones found at this resolution. 'Too close' is defined as a quarter of the chord length. At successively decreasing chord lengths critical points may be found between those determined at coarser resolutions. The various analyses are carried out independently, in the sense that partition points found at different scales were no expected to support each other.

The algorithm therefore finds corners at different resolutions. However, the resulting description of the curve structure fails to satisfy the requirements for an analysis scheme to identify possible 'faces', as detailed above. It could serve as a possible starting point, as will be shown below [5.3.1].

Fischler and Bolles describe a further algorithm for curve partitioning based on the idea that the curve is the product of various underlying 'processes' which create segments of different character; the partitioning scheme is viewed as one of detecting process homogeneity. However, this again is inadequate since it operates only in the relatively simple environment of noiseless curves composed exclusively of straight lines and arcs of circles.

Phillips and Rosenfeld [1987] presented a method of curve partitioning using the arc-chord distance which is similar to that of Fischler and Bolles. The method appeared to give comparable results, but was computationally and conceptually simpler.

The method uses a constant arc length k, length being measured in chaincode chainlets, in contrast to Fischler and Bolles' use of a constant chord length. For each point P on the curve the distance d(P,C) is found, which is the perpendicular distance from P to a chord C whose arc has length k and has P in its interior. There will be k+1 such arcs, two of which have P as endpoint and so will give d(P,C) values of zero. Let M(P,C) be the maximum of the d(P,C) values, for all such chords. The points selected as critical (i.e. corner) points are those points P for which M(P,C) is a local maximum. Phillips and Rosenfeld also required the M(P,C) value to exceed a set threshold t, which is a function of k. They used $t = k/5$, which is approximately $(k/2).\cos(3\pi/8)$ and so is the altitude of an isosceles triangle whose vertex angle is $3\pi/4$ and whose equal sides have lengths $k/2$.

This completes their published description of the corner selection method. However, some additional processing appears to have been used, as can be seen from a comparison of the published graphs of M(P,C) and the corresponding selection of vertices on the contours. The larger peaks in M(P,C) are often 'crenellated', i.e. have many smaller peaks superimposed. These minor peaks are also local maxima, and so would be presented as candidate corners by the algorithm as described, but the display of corner points show that these 'false' local maxima have been somehow eliminated.

Noting that the success of their method depends on the choice of appropriate arc lengths, Phillips and Rosenfeld describe a possible method to achieve this. They determine the best fit straight line for each k-arc of the contour, and compute the RMS error corresponding to this fit. This process is repeated for a sequence of arc lengths, producing a sequence of fit measures for each contour point. In a given part of the contour 'good' values of k are taken as those which produce local minima in the fit measure. Apart from suggesting a scheme for efficient implementation of the calculations involved, the idea is not developed further.

5.2.2.4 Lowe's algorithms

Lowe and Binford [1983] developed an algorithm for curve analysis which detected significant structure at multiple resolutions. The term 'significant structure' meant perceptually significant collinear or curvilinear segments. Details of the curvilinear analysis were not given in the paper; Lowe later re-addressed this problem [Lowe 1988], building on the work of Mokhtarian and Mackworth [1986] cited above.

The analysis of collinear segments attempted to measure the perceptual significance of all possible groupings. The significance measure used the idea of nonrandomness in linearity, defined as how unlikely a point is to be as close as it is to a curve, given its distance from the closest defining point of the curve.

Not every possible segment of a curve was tested for nonrandomness. Groupings at all scales differing by factors of two were examined, from groupings of three adjacent points up to groupings covering the full length of curve. At each scale, grouping were examined which overlapped by 50%, which meant that any given segment of the curve had at least one grouping attempted which covered half its length but did not extend beyond its borders. As a consequence of this strategy there is no guarantee that the maximal extent of straight portions would be found.

After measuring the significance of each grouping, a procedure was executed which stepped through the different resolutions at each location along the curve, selecting only those segmentations which were locally maximum in their significance values. An absolute threshold of significance was also used. There could be more than one local maximum if the curve exhibited different structure at different resolutions of grouping. Overlapping segments could also result.

The measure of significance of curve segments based on nonrandomness in linearity was dropped by Lowe in later work, and replaced by the aspect ratio of a covering rectangular strip [Lowe 1987]. The resulting algorithm was based on the strip tree representation of curves.

Strip Trees

Ballard [1981] introduced the idea of the 'strip tree': a hierarchical data structure for the representation of curves. The strip tree is a binary tree where each node represents a 'strip segment', a rectangle covering a segment of the curve. Figure 5.5 illustrates the principle: the orientation of the rectangle covering a segment is determined by the slope of the line joining the endpoints, and the width of the rectangle is then the minimum value which provides a cover for the contour segment

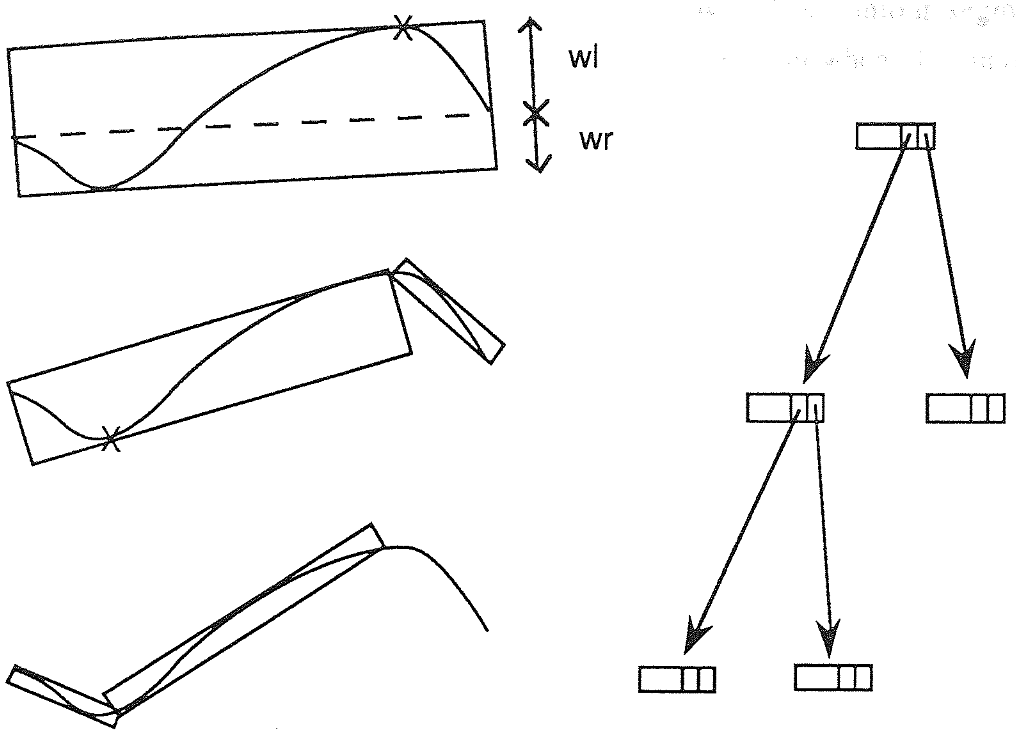


Figure 5.5 Strip segments in a strip tree.

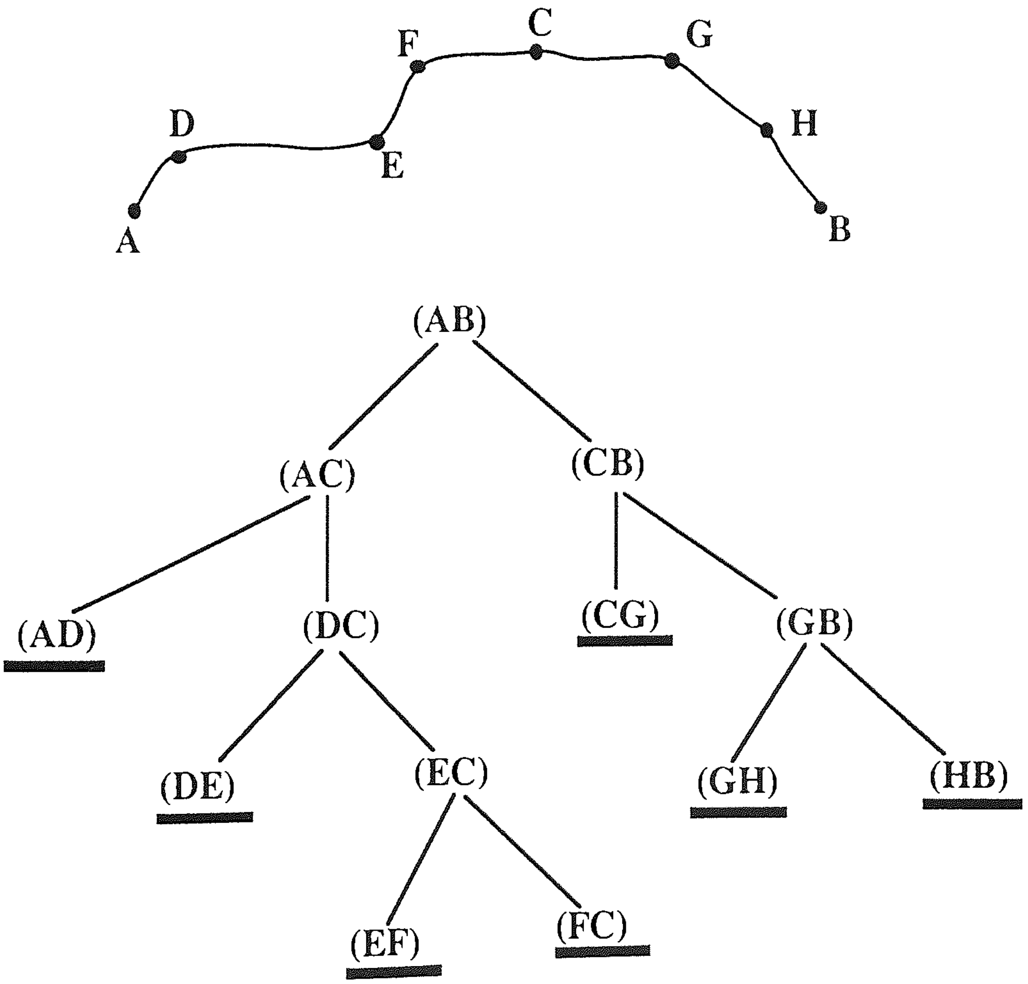


Figure 5.6 A curve segment and its corresponding strip tree

Each non-leaf node in the strip tree has two sons. The contour segment represented at the father is partitioned into two pieces, at the point where the curve is furthest from the straight line segment connecting the endpoints, and the two sons represent approximations to these pieces. This subdivision process terminates when each curve segment is sufficiently straight, and the corresponding strip segment sufficiently narrow. Figure 5.6 illustrates a curve portion and an outline of its corresponding strip tree.

Each node in the strip tree contains the locations of the beginning and end points of the contour segment represented by this strip ($\underline{x}$b and $\underline{x}$e) and the width of the strip segment, as two half-widths: width = wr+wl. Lowe extended this representation by storing in addition a significance measure for each strip, determined by the ratio of the line segment length divided by the maximum deviation of the curve from the straight line segment.

The algorithm suggested by Lowe [1987] builds up the tree structure in a recursive procedure, halting when each strip has reached prescribed limits on its length or width. As the recursion unwinds each node passes its significance information up the tree, so that if the significance of a segment compares favourably with that of any of its subsegments then the subsegmentation is ignored, removing a subtree from the strip tree.

This algorithm was implemented using only slight modifications, some of which have also been suggested by Rosin and West [1988]. The recursion was terminated when the strip length became less than 4 pixels or the strip width became less than 3 pixels. In order to avoid divide-by-zero errors the significance measure used by Lowe, of strip length divided by strip width, was replaced by its inverse. To remove possible ambiguity, from now on it is this quotient which will be meant by the 'aspect ratio' of a strip: width divided by length. Thus the lower the aspect ratio, the more significant the line.

The algorithm effectively analyses the contour at various resolutions and chooses the single polygonal approximation which contains the most perceptually significant straight segments represented in the tree structure. However, the analyses do not consider all possible segments, only those generated by the recursive splitting process. The process is scale independent, and avoids the use of any arbitrary error thresholds.

Figure 5.7 provides an example of this algorithm's operation. Figure 5.7(a) shows a test contour, and 5.7(b) the polygonal approximation produced. It is considered that this is not the preferred result, since a longer irregular 'face' has been found to be more significant than any of its subfaces. Introducing a small bias into the calculation can force the algorithm to deliver these subfaces; the resulting polygonal approximation is shown in figure 5.7(c). However, this example also illustrates another weakness of the algorithm: the merging process operates by collapsing complete subtrees, and there is no opportunity to merge segments which are represented in two different subtrees. Thus the left hand face is found as two subfaces and the algorithm gives no opportunity for these subfaces to be merged.

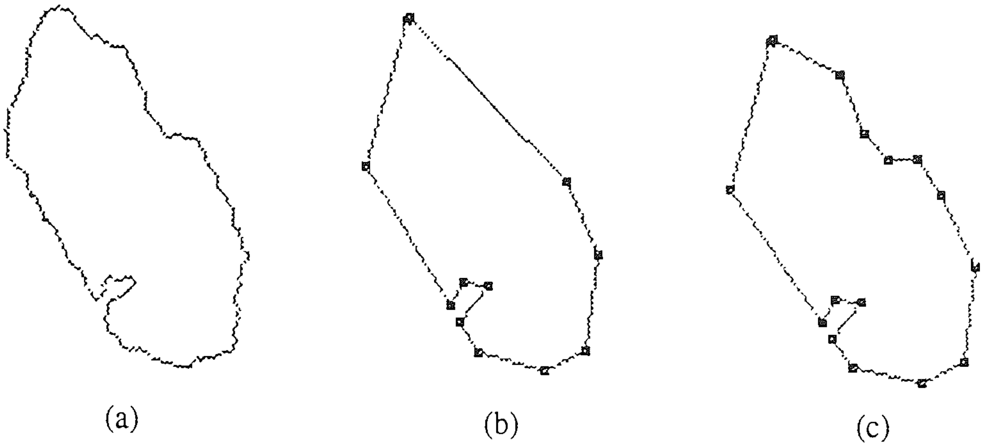


Figure 5.7 A contour and two polygonal approximations created via strip trees

This effect can also be seen in figure 5.6: the first subdivision of segment AB is made at point C, which lies on a minor protrusion on a relatively straight portion of the curve. As a result, segments FC and CG lie in different subtrees and cannot be merged to produce the significant segment FG.

Some change to the merge phase therefore appears to be necessary, so that possible merges of adjacent faces can be considered regardless of their position in the tree. The sort of merge procedure used in the split and merge algorithm would seem to be appropriate. Such a change would mean the loss of the binary tree structure describing the hierarchy of faces.

In order to satisfy the requirements for crystal profile analysis, discussed above, the algorithm would also need to be modified to find faces which, on the basis of straightness alone, might be considered sub-optimal. The final selection of faces may then be made by higher level routines which can use additional global information such as symmetry and collinearity. These features will be incorporated in the new algorithm described below [5.3].

5.2.2.5 The work of Bengtson and Eklundh

This review of publications on multiple resolution contour analysis concludes with a consideration of the work of Bengtson. This recently published material was encountered too late to influence any part of the development of the analysis scheme described below; the similarities which can be found stem from a similar motivation and common influences.

Bengtson and Eklundh [1989] consider the problem of deriving robust descriptions of planar contours which preserve essential shape features. A multiscale analysis is used. Their work builds on earlier research published only as a technical report at the Royal Institute of Technology, Stockholm [Bengtson *et al.* 1986]. The motivation for the work reflects the comments of chapter 4 on deriving descriptions of geometric structure for intermediate level computer vision, and the importance of contour cues in recognising occlusion. A contour representation in the form of a 'segment tree' is developed, influenced by Witkin's ideas of scale-space and the 'interval tree' [5.2.2.1].

The algorithm proceeds in stages. First a family of polygonal approximations in the form of a tree structure is produced, describing the contour at various resolutions. Information about where the contour is straight and curved can be derived from this tree. The tree is used to construct a new polygon in which curved segments are represented with a greater number of points, thus indicating the curved shape better. Finally the new polygon is used as input to an algorithm which produces an approximation of the contour as straight lines and conic sections.

In the first stage the family of polygonal approximations ranging from coarse to fine is produced using what is, in principle, a split-and-merge algorithm at each level. There is, however, a two-way communication between levels. The result at one level serves as an input to the next, coarser level and the position of breakpoints is determined by the finer level. A new breakpoint can be introduced at a coarser level and if there is no corresponding point at finer levels such a point is inserted into the approximations. This results in a nested set of approximations, forming a tree structure of segments. It is possible to use this representation to generate the polygonal approximation which is most stable or scale invariant over the current range of scales.

The split-and-merge algorithm has a 'hop-along' implementation: given a start position the algorithm jumps to a point some distance along the curve and then tests whether a line fits this segment. If so, a new step is taken along the contour and the test repeated. If this 'split test' is not passed a breakpoint is introduced along the segment at the point of maximum deviation from the chord, and the procedure is repeated for the two subsegments obtained. Since this hop-along stategy cannot allow line segments longer than the step size a merge step is added which tests if consecutive line segments are almost collinear. This is the 'merge test'.

The Bengtson algorithm uses a collinearity test of the form $e \leq E$ for the split decision. The error e is the distance to the chord and not to the line of which the chord is a segment, which can lead to the more accurate siting of breakpoints in the splitting phase. The merge test is of the form $|A/L| \leq E/2$, where A is the area between the arc and the chord, and L is the arc length. The same error threshold E is used in both tests. In addition, the hop-along step size is adaptively modified to reflect the behaviour of the curve. The stepsize increases after a successful collinearity test and reduces when a curved portion is encountered.

The multiscale approach uses a set of error thresholds $E^{(s)} = s\Delta E$, s=0,..,S. The requirements of multiscale analysis mean it is not sufficent to implement the split and merge rules outlined above in a straightforward manner: some linkage between levels must be introduced. Modifications are introduced into the split phase. Suppose the finer polygonal approximation exists and the split phase is underway with the last breakpoint found at q. As described above, the (adaptive) hop-along step size is used to produce a segment which is submitted to the split test. Testing and splitting may proceed recursively until a segment is produced which satisfies the split test. This results in a suggested breakpoint at p, say. An attempt is then made to find a breakpoint from the finer approximation in the neighbourhood of p which also satisfies the split test. Points are sought first beyond p, and if this search is unsuccessful, between p and q. If no suitable breakpoint is found in the finer approximation p itself is chosen, and inserted into all previous polygonal approximations. The signed area of the triangle formed by the last three breakpoints is then calculated, and saved if it is less than E. The accumulated area is used in the $|A/L|$ parameter of the final merge phase. The final polygonal approximation is the result required for this resolution, although it may possibly be modified by the insertion of additional breakpoints created at a coarser level.

A polygon in the tree produced in this way must be modified to make it suitable for input to the procedure which will create the first and second order approximation. This modification will make use of the subset of all polygons representing finer tolerances. Any of the polygons from the tree could be selected as a basis for the approximation, but for perceptually satisfactory results the polygon chosen should be in some sense stable. This reflects the principle that the characteristic features of a signal are those which are stable over a range of scales.

Choosing the best polygon is equivalent to choosing the error threshold used in its creation. The stable features sought are the number of vertices and number of inflections on the polygon, i.e. segments with a local concavity at one end and a local convexity at the other. The numbers of these features in the polygonal approximations are examined as the tolerance is increased, and tolerances sought which produce relatively little change.

It must be noted that the assumption is used that there exists at least one scale interval where all parts of the contour are stable. This may easily not be the case: the optimal scale may be different for different parts of the contour. A solution to this problem is still being sought.

With the stable scale selected, corner points are found and the contour segmented into straight and curved parts. This process uses the set of all polygonal approximations with smaller error tolerances. Extra vertices are inserted to better approximate curved parts: curved parts are essentially defined as those sections of the contour where vertices are close together. Points separating straight parts are corners, as are points with sharp vertex angles. Within the curved parts of the contour the positions of inflection are found. The tangent values at corners and inflection points are then estimated. Finally each curved part of the contour is approximated with conic splines. Bengtson has not considered introducing non-local symmetry information into the analysis.

The analysis scheme, still being developed by Bengtson, appears to provide the information required for finding faces on crystal profiles. The additional description of curved, anhedral portions of the contour could also be valuable. The outstanding problem, however, is the assumption that a single resolution can be found which is appropriate for all portions of the contour. The multiscale analysis algorithm which will now be described offers a solution to the problem of variation in optimal scale around the contour.

5.3 A new algorithm for multiscale contour analysis

A new algorithm has been developed in order to find candidate faces on a crystal cross-section image. The algorithm seeks significantly straight portions of the contour, operating at a range of scales. The various levels of structure are presented in a hierarchical form, in what is termed the face tree. The algorithm could be viewed as an extension of the approach used by Lowe, described above [5.2.2.3], although ideas seen in the work of Davis, Witkin, and Phillips and Rosenfeld, also discussed above, have been exploited. The work also has some similarities with the approach used by Bengtson, arising from similar motivation and common influences. In contrast with Bengtson's algorithm the new algorithm does not assume that one single resolution can be found which is appropriate for the complete contour.

Unlike the strip tree approach used by Lowe, the algorithm can offer multiple representations for portions of the contour when appropriate. As discussed above, when more than one possible level of structure can be discerned the final choice of faces will be decided by later higher level analyses which use properties of the figure such as symmetry or collinearity. In this way non-local information will be brought into the face finding process and the final analysis will not rely purely on local measures of straightness on the contour.

A description of the possible faces found can be produced in the form of a tree structure. Again in contrast with the strip tree method, this tree is unrestricted in the order of its nodes (i.e. degree of branching). Associated with each (possible) face found by the algorithm is a list of properties required in later analyses such as symmetry appraisal or the evaluation of quality of development of faces. These attributes include length, aspect ratio of a covering rectangular strip, start and end points on the contour, and direction.

The algorithm also offers a 'conventional' polygonal approximation, so providing a full description of the structure of the entire contour, and not only of the faces which may be present. Using a fixed error of fit measure, a complete approximation of the contour can be produced, for purposes of data compression. This is important for improving the efficiency of subsequent calculations such as the creation of convex hulls and measurement of area, when the faces found do not provide a suitable approximation for the whole contour.

The fixed error polygonal approximation is derived from the same intermediate representation as is used in the face-finding calculations. Compatibility of vertex positions between these two representations of the contour is maintained. As a result, for any set of faces selected from the proffered tree structure, a complementary polyline approximation is available for those portions of the contour which have not been nominated as faces, so providing a description of the complete contour.

The complete algorithm will now be described in detail. First the corner-finding technique of Phillips and Rosenfeld is extended to create a new dual intermediate representation of the contour. This representation is the starting point for the subsequent processing for finding likely faces on the contour and constructing quantitative descriptions of their form and hierarchical relations: candidate faces are found via their corresponding characteristic 'zones' in the intermediate representation, and the face tree for the contour is created from the set of candidate faces.

5.3.1 Creating an intermediate representation for contours

The arc-chord calculations used by Phillips and Rosenfeld, discussed above [5.2.2.3], have been adapted to yield much more information describing the contour. In particular, their work has been extended to produce a new intermediate representation of the structure of the contour. Before describing the method and the representation some remarks on input data assumptions will be given, and some notation introduced.

The contour is presented to the algorithm in standard chaincoded form. The x–y coordinates of the (arbitrary) start position are also given, which then permits the calculation of the Cartesian coordinates of every point on the contour. Length along the contour is measured in numbers of chainlets, which is only an approximation to the true value, in general. Points on the contour may be referred to simply by the number of the chainlet at which the point is reached, when following the coded boundary from the start position.

The Phillips algorithm requires, for each point on the contour, the calculation of the perpendicular distance from the point to each member of a set of chords whose arcs all have the same length and which contain the point. Let the perimeter length be L, and the arc length k. Then one may imagine chords drawn from every contour point to a point k steps further along the contour. For the arbitrary point p on the contour, an arc and chord extend from p to (p+k), where this latter value may be taken modulo L, corresponding to 'wraparound' on the closed contour.

For each point on the arc the perpendicular distance to the chord is found (more precisely, the distance to the line of which the chord is a segment). Let e(p,m,k) denote the distance from contour point (p+m) to the chord from p to p+k; e(p,0,k) = 0, and e(p,k,k) = 0, and need not be calculated. For each arc length value k, the e(p,m,k) values provide a two-dimensional array of distance values, of L columns and (k−1) rows. This may be termed E(k):

$$E(k) = \{ e(p,m,k) \mid 0 \le p < L \text{ and } 0 < m < k \}$$

The e(p,m,k) values, for $0 \le p < L$, $0 < m < k$ and $k_{min} \le k \le k_{max}$, constitute a three-dimensional 'wedge' of distance values, which is being considered one two-dimensional section at a time, i.e. for each particular arc length k.

Two useful measures can be extracted by relatively simple examination of the array E(k). For each chord the maximum arc-chord distance can be found; for each contour point the maximum arc-chord distance from the point to all chords of k−arcs which contain the point can be found. Care must be taken to handle the 'wraparound' when analysing the E(k) values: this two-dimensional array may be more accurately thought of as being cylindrical rather than rectangular.

The maximum arc-chord separation for the arc and chord from point p to (p+k) may be termed maxex(p,k):

$$\text{maxex}(p,k) = \max\{ e(p,m,k) \mid 0<m<k \}.$$

These values may be extracted from the E(k) array by finding the maximum value in column p of the array.

Finding the maximum arc-chord distance from a point p to all chords of arcs which contain p, which is the basis of the Phillips algorithm, is a matter of finding the maximum along an oblique line through the array of E(k) values. For the contour point p, this quantity will be termed emax(p,k):

$$\text{emax}(p,k) = \max\{ e(p-m,m,k) \mid 0<m<k \}.$$

Thus for each arc length k a two-dimensional array of e(p,m,k) values is created, from which two linear arrays are constructed: maxex() and emax(). It is important to note that the calculations for the construction of these two linear arrays require one row only of the E(k) array to be available at any one time during computation. It is not necessary to store the complete, and possibly very large, array of values.

These calculations are repeated for a range of arc length values k. The sets of one-dimensional arrays of emax(p,k) and maxex(p,k) values generated over all arc lengths constitute two two–dimensional matrices. In the implementation of this scheme the two-dimensional array of emax(p,k) values is referred to as 'cpspace' ('corner-point scale-space') the array of maxex(p,k) values is termed 'acspace' ('arc-chord scale-space'). The renaming reflects the fact that these arrays can be used to describe important features of the contour over changes in resolution (i.e. arc length).

Locations in these arrays are specified by two coordinates: the a–coordinate which represents the arc length used, and the p-coordinate which represents the position on the contour. The convention used in presenting these arrays is to use p as the abscissa and a as the ordinate.

The cpspace values provide a record of the curvature along the contour, measured over a range of resolutions. This matrix is therefore very similar to the intermediate representation produced by Davis' ADS algorithm [5.2.2.2], and Davis' techniques could be adapted to analyse this information. As will now be described, the acspace values can be processed further to yield 'dual' information regarding the sides present in the contour. These two representations can then be used in conjunction to analyse the form of the contour. In contrast to the representations used in Davis' side and angle analyses, these have a common origin. This factor facilitates their combined use.

The generation of the 'straightness' information, which will complement the corner information, is based on the observation that for any contour point p it is possible to judge whether it lies on a straight portion of the contour by first finding the minimum of the maximum arc-chord distances (i.e. minimum maxex value) for all the arcs which contain the point, and then comparing this value with the arc length or the chord length. This minimum maximum arc-chord distance is similar to the minimax error criterion used by Kurozumi and Davis [1982]; one difference is the additional constraint that the endpoints of the approximating straight line segment used are required to lie on the original contour.

By using a small modification to these calculations a measure of fit can be obtained which is identical to that used in the strip tree calculations described above [5.2.2.4]. Instead of storing only the maximum absolute arc-chord separation for each arc-chord pair, account is taken of the sign of the deviation and the maximum deflection of the chord to the right and to the left are both stored, i.e. the wr and wl values defined for the strip tree representation. The measure of arc-chord separation used is the sum of these two widths. This value can be seen to be the width of the covering rectangular strip for the arc, whose direction is parallel to the chord. It is therefore identical to the strip width used in the strip tree algorithm.

When this modification has been used in the calculation of the maxex values the required new array of values can be produced. This array is termed 'mmspace' ('minimax scale-space'). If row a of the acspace array represents strip width values calculated using an arclength k, then the corresponding mmspace values are found by searching the appropriate neighbourhood for the minimum value:

$$\text{mmspace}(p,a) = \min\{\ \text{acspace}(p-m,a) \mid 0 \leq m \leq k\ \}.$$

The two core arrays of the algorithm have now been produced: mmspace and cpspace. These arrays together constitute the intermediate representation which is the basis of further processing. This dual representation of the contour describes the structure over a range of resolutions; mmspace holds information on the 'sides' in the contour and cpspace holds information on 'corners'. The task of the algorithm is to use these complementary sources of information to extract the required structural descriptions.

5.3.2 An outline of the intermediate representation

An examination will now be made of how some contour features appear in the intermediate representation. This will lead to a consideration of how the key features can be identified and their descriptions extracted. The analysis is based primarily on mmspace, reflecting the relative importance of linear features over corners, but first the properties of the cpspace will be considered.

5.3.2.1 'cpspace'

Each row of the cpspace array represents the corner measures found around the contour using Phillips' method for one particular arc length value. Looking at the array values as a surface in three dimensions, ridges along lines of near constant p coordinate mark corners on the profile at these position. The relative prominence of a ridge varies with arc length; ridges disappear into the surface as the arc length grows beyond the size of the domains of the corresponding angles.

Techniques similar to those developed in ADS could be used to move over this surface gathering information across different resolution to identify the key points on the ridges which would represent the desired set of corners. However, no such analysis is necessary, since the mmspace information can be exploited. The only processing which may be carried out is the marking of the ridges, at all resolutions.

In order to assist corner-finding the algorithm marks local maxima and plateaux in cpspace by placing the value 1 in a corresponding array. Minor 'insignificant' peaks in cpspace are local maxima and so are also marked. However, this is not a problem since, as will be shown, information derived from the mmspace array is used to position search windows on cpspace so that the highest peaks found are significant corners and not minor corrugations.

5.3.2.2 'mmspace'

As a first step towards understanding the properties of the mmspace representation of contours some simple, noiseless polygonal contours will be considered. Clearly only a simple corner finder is required to define the structure when such figures are being analysed, and so only an examination of the cpspace array would be necessary; the following examples are not put forward as recommended processing schemes for such figures but as illustrations of the nature of mmspace.

It should be noted that references to portions of a closed contour by giving their endpoints are made unambiguous by using an implicit clockwise ordering of points. Thus, in discussing a polygonal contour ABCD, 'arc AC' refers to arc ABC and not ADC. A possible source of confusion may be the use of two definitions for length on the contour: arc length and what will be termed 'pseudolength'. The pseudolength of a contour portion may be defined as the number of points it contains. Then a segment of pseudolength n corresponds to an arc length of n−1, since only n−1 chainlets are required to proceed from start to endpoint. The exception is the entire, closed contour, for which both definitions of length agree. The length of a segment AB may be represented as |AB|; this will be used to indicate arc length only.

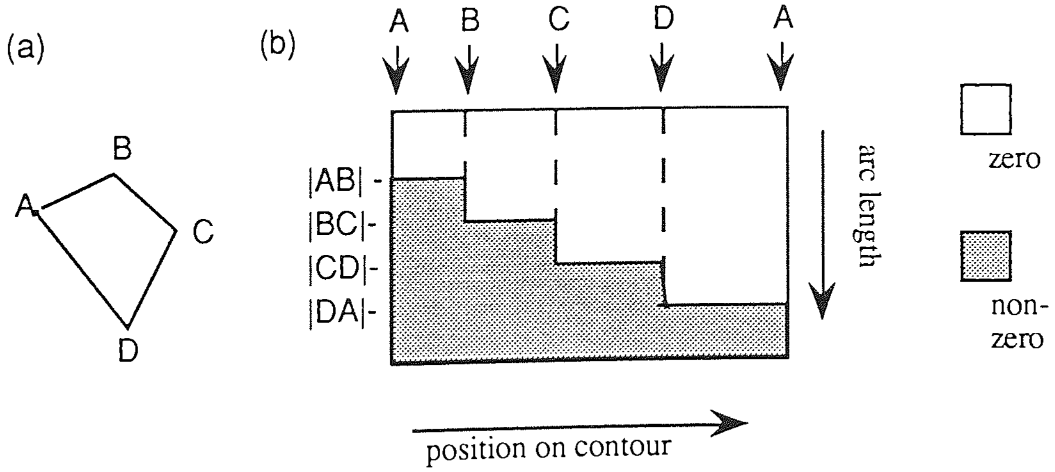


Figure 5.8 Sketch of 'mmspace' representation of an irregular quadrilateral

First a simple quadrilateral polygon ABDC will be considered, with sides of different lengths: |AB|<|BC|<|CD|<|DA|. This is illustrated in figure 5.8(a). It is clear that every point on the contour lies on a perfectly straight line segment of length at least |AB|. Therefore, for every contour point, any arc length not exceeding |AB| will produce an mmspace value of zero.

When the arc length k is greater than |AB| and less than or equal to |BC| the minimum strip width which can cover a portion of the contour which has length k and which contains a point on side AB is no longer zero. Thus, for points on a polygon face, the mmspace values will rise with increasing arc length, once this length exceeds that of the face. It should be noted that points A and B, as corners, lie on two faces, and so mmspace values at B will continue to be zero until the arc length exceeds |BC|, and at A they will remain zero until the length exceeds |DA|. Figure 5.8(b) gives a sketch of the behaviour of the mmspace values; the behaviour of non-zero values is not indicated.

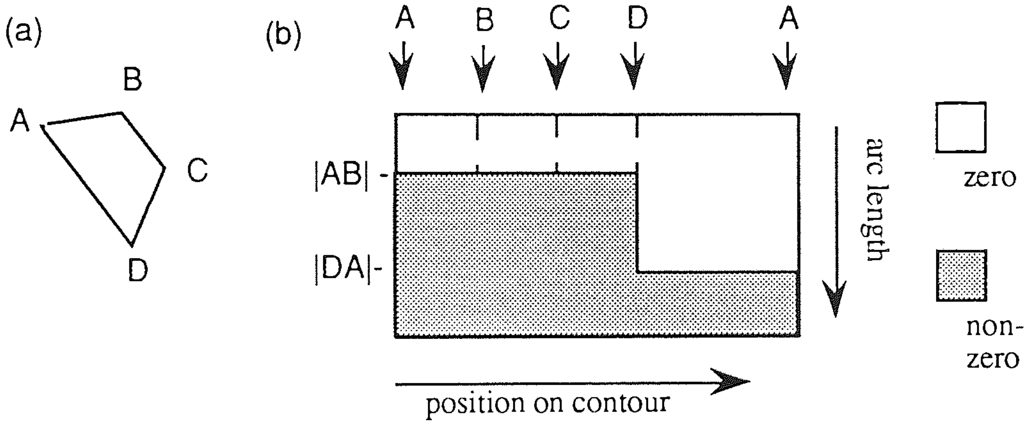


Figure 5.9 Sketch of 'mmspace' representation for a semi-regular polygon.

Figure 5.9 shows the situation when adjacent pairs of sides of a polygon have the same length: here |AB| = |BC| = |CD|. In contrast to figure 5.8, where all the corners could be inferred by the 'steps' seen in mmspace, here mmspace does not immediately signal the presence of corners B and C. However, their existence can be inferred from mmspace alone, and in some situations their positions can also be deduced.

In contrast to figure 5.8, figure 5.9 shows only two 'steps' in mmspace, i.e. line segments marking the onset of higher mmspace values as arc lengths increase. For contour points lying strictly between A and D the mmspace values rise above zero for arc lengths greater than |AB|, implying that all these points lie on faces whose lengths do not exceed that value. However, the step which marks the last arc not exceeding |AB| in length, extends to a length of |AB|+|BC|+|CD|. If, for example, the last arc length used were exactly |AB| and the arc lengths successively doubled as the rows of mmspace were traversed then the next arc length would be 2|AB|, and would produce a strip width of zero for points on faces up to this value in length. The fact that a nonzero strip width has been found implies that the portion of contour from A to D must consist of faces whose lengths are at least |AB| and strictly less than 2|AB|.

Further, the corners must lie at least the distance |AB| from A and from D (otherwise the arc length |AB| would not have produced a zero strip width for the segments AB and CD). This implies that contour segment AD must consist either of three faces of identical length, or of two faces, with their junction somewhere in the middle third of the segment. If the arc lengths were increasing more slowly, say only one chainlet at a time, one could deduce that the former situation must apply, since no straight contour portions of length |AB|+1 were found within contour segment AD. This explains the above remark that in some situations an analysis of mmspace could yield the position of polygon corners.

To see how the cpspace values can be used a specific example may be considered. The series of arc lengths assumed will be the doubling progression 2,4,8,16... The pseudolengths of AB, BC, and CD will be taken as 25 (i.e. spanned by an arc length of 24) and that of DC as 35. Arc lengths 2, 4, 8 and 16 will yield mmspace values of zero at all contour positions, and a length of 32 will yield values of zero only for contour positions from D to A inclusive.

As explained above, an examination of the lengths of the zero to non-zero steps in mmspace reveals that contour segment AD must contain one or two corners. Further, the corner(s) must lie between A+16 and D-16 (since otherwise a zero strip width would not have been found using arc length 16 between A and A+16 and between D-16 and D). The corresponding cpspace values are therefore examined, and the maximum value within this window is selected. The maximum will lie either at B or C; it may be assumed it is found at B. The zero mmspace value found for arclength 16 is again exploited by using the implication that no corner can lie within the span of this arc length from B. This reduces the search window to a small zone from B+16 to D-16. Taking the maximum cpspace value in this window locates the corner at C. Repeating this process by restricting the search window in the neighbourhood of C now collapses it completely. The search therefore terminates successfully, having found the corners at B and C.

In this way knowledge of the maximum arc length which indicated the presence of a face can be used to set up a search window on the cpspace values, and also to suppress the search around any corners subsequently identified in that window. This principle can be extended and used in the analysis of general contours, and not only ideal polygons.

5.3.3 Analysis of the intermediate representation

The mmspace array describes how the minimum width of covering strips change as the length of the strips increase. Even when an mmspace representation is generated for a noisy, non-ideal polygon, where covering strips which are not very short have nonzero widths, it is possible to identify faces. This is achieved in two stages. First 'stable' portions of the contour are found, in a process corresponding to finding segments AD and DA of the ideal polygon of figure 5.9(a), for example. These portions, which will be termed 'zones', are then analysed to see if they represent a single face or a set of two or more faces. As in the analysis of figure 5.9 described above, the faces in a zone should have approximately similar lengths.

The notion of 'stability' of regions in mmspace requires explanation and comment. The key observation is that relatively straight segments of a curve are characterised by comparatively small changes in the width of the minimal covering strip as the length of the strip is increased. This situation corresponds to a relatively 'flat' rectangular area in mmspace defining the extent of the segment, and whose stability, i.e. range of scale covered, reflects its perceptual significance. This observation may be compared with that of Witkin [1983], who noted that stable rectangles in scale space, i.e. those which extended over a relatively broad range of scale changes, corresponded to intervals in the signal which tended to "leap out at the eye".

The final aim of the analysis algorithm can be described as locating a set of line segments in mmspace. Each line segment, termed a 'face marker', represents a possible face on the contour and carries much implicit information about that face simply by its position in mmspace. The example of figure 5.10 may clarify the situation. Figure 5.10(a) gives an example of a irregular contour. The contour can be seen to be approximately pentagonal with sides which are non-uniform in character and some of which show more than one level of linear structure. The origin of the chaincode is near the base of the irregular, vertical right side. The contour has been encoded in a clockwise direction. There are 483 chainlets in the encoded contour.

Figure 5.10(b) shows the mmspace representation of this contour, with higher values indicated by the darker grey tones. The axes are shown on the corresponding 'analysis' in figure 5.10(c), where p-coordinate values have also been marked. 15 different arc lengths, ranging from 2 to 307, were used to generate the mmspace values.

Two large zero-valued rectangles are evident in mmspace, corresponding to the two perfectly straight sections of the contour extending from points 105 to 230 and from 365 to 420 (all such positional values are approximate). Other sides give rise to rectangles of relatively constant non-zero values. The contour segment from 420 to 470, for example, which forms a noisy oblique face on the top left of the figure, gives a relatively clear rectangle in mmspace, extending to an a-coordinate which corresponds to an arc length value of 40.

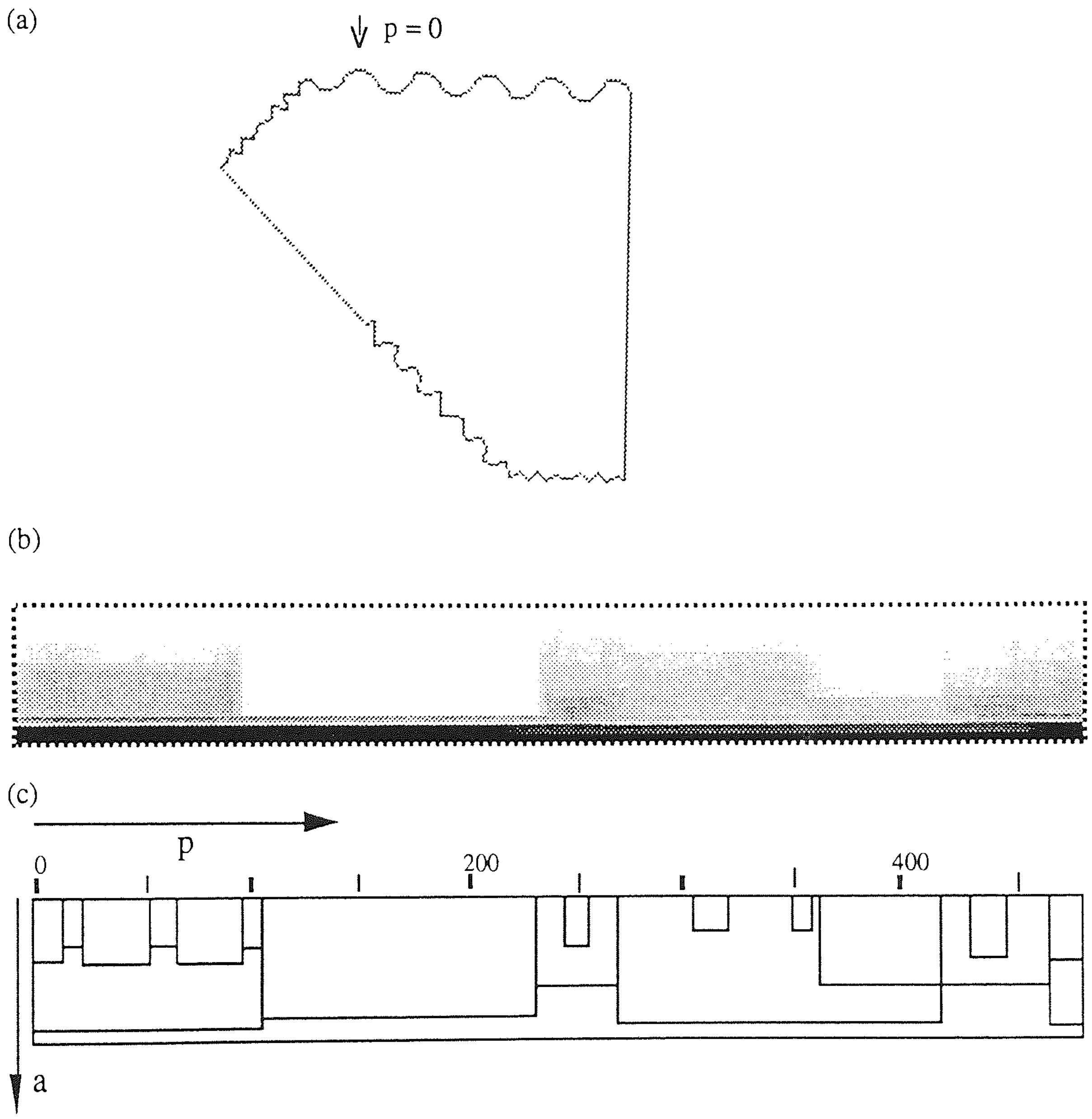


Figure 5.10 An irregular contour and a sketch of its mmspace representation

The different levels of linear structure present on the contour are reflected in mmspace by the arrangement of nested rectangles. This has been indicated in the analysis of figure 5.10(c). For example, the relatively smooth central portion of the short face on the bottom right of the figure gives a lighter toned grey rectangle, extending from p-coordinate 245 to 255, within the rectangle representing the whole face. However, rectangles corresponding to the finer resolution structure of the face along the top of the figure cannot all be seen: only six rectangles are marked in the analysis, nested within the larger rectangle which extends from 470 to 105. One of these six rectangles 'wraps around' from p-coordinates 470 to 15. The larger rectangle represents the whole face; the contained rectangles mark the *zones* of relatively constant character, and the larger of these represent pairs of subfaces. This example of a step representing multiple faces corresponds to the situation discussed in relation to figure 5.9. The rectangles corresponding to faces (c.f. zones) are marked in figure 5.11 below. Such an analysis requires the use of the cpspace corner information.

The major oblique face on the left side of the contour shows two parts of different character. The whole face is represented by the rectangle which extends from p–coordinate 270 to 420; the smoother subsegment by the zero-valued rectangle nested within. Smaller low-valued rectangles between p–values 270 and 365 reflect linear structure within the irregularities. The small rectangle from 305 to 315 represents linear structure found using arc lengths up to the value 5: this corresponds to two short but quite straight segments within the noisy half of this face. Again, the small rectangle is actually two smaller rectangles side by side; an examination of cpspace (not shown) confirms this.

It can be seen that the lower edges of the stable rectangles in mmspace carry all the information represented by the rectangles. These line segments will be termed 'zone markers' since they mark and describe stable zones on the contour. They also carry information regarding the resolutions at which this structure is stable. The p–coordinates of a zone marker define the extent of a stable section of the contour. The a–coordinate of the line segment defines an arc length c, and the mmspace value sampled on this line segment gives a width w, which together describe the dimensions of rectangular strips which can be used to cover the contour section: any point within this section of the contour lies within some subsection spanned by an arc length c which has a corresponding strip width not exceeding w. This arc length need not be the longest possible value which could achieve this, but it is the longest of the set of arc lengths tested. It may be noted that this width constraint is not saying that a rectangular covering strip of the given dimensions can be 'slid along' the contour section; the section may contain corners.

As mentioned above, the second phase of the analysis is to check stable contour segments to see if they represent more than one face. This can be viewed as a check on each stable rectangle found in mmspace to see if it is actually two or more abutting rectangles of the same height and strip width value. This will require the use of the cpspace information. Since the essential information about a rectangle is held in the lower edge zone marker the check takes the form of an examination of zone markers to see if they are single 'face markers', or a concatenation of two or more.

Thus the result of the second analysis phase is a set of face markers in mmspace. Associated with each of these line segments in mmspace is a set of attributes which can be translated into descriptions of the corresponding face on the contour. The p-coordinates of the line segment define the position and length of the face. As for the zone marker, the a-coordinate of the segment and the mmspace value on it together describe the dimensions of an optimal covering strip for the face.

The arrangement of face markers in mmspace can be viewed as a quantitative description of the multiresolution structure of the contour. The nature of the various levels of linear structure around the contour is represented by the mutual arrangement of the face markers in mmspace. Projecting them on to the p-axis can reveal how many levels of structure are present on any portion of the contour. The separation in the a-direction of two markers reflects the difference in the resolutions of the corresponding faces. In particular, the difference in a-coordinates between a face marker and the markers of its subfaces indicates the 'uniqueness' of the face.

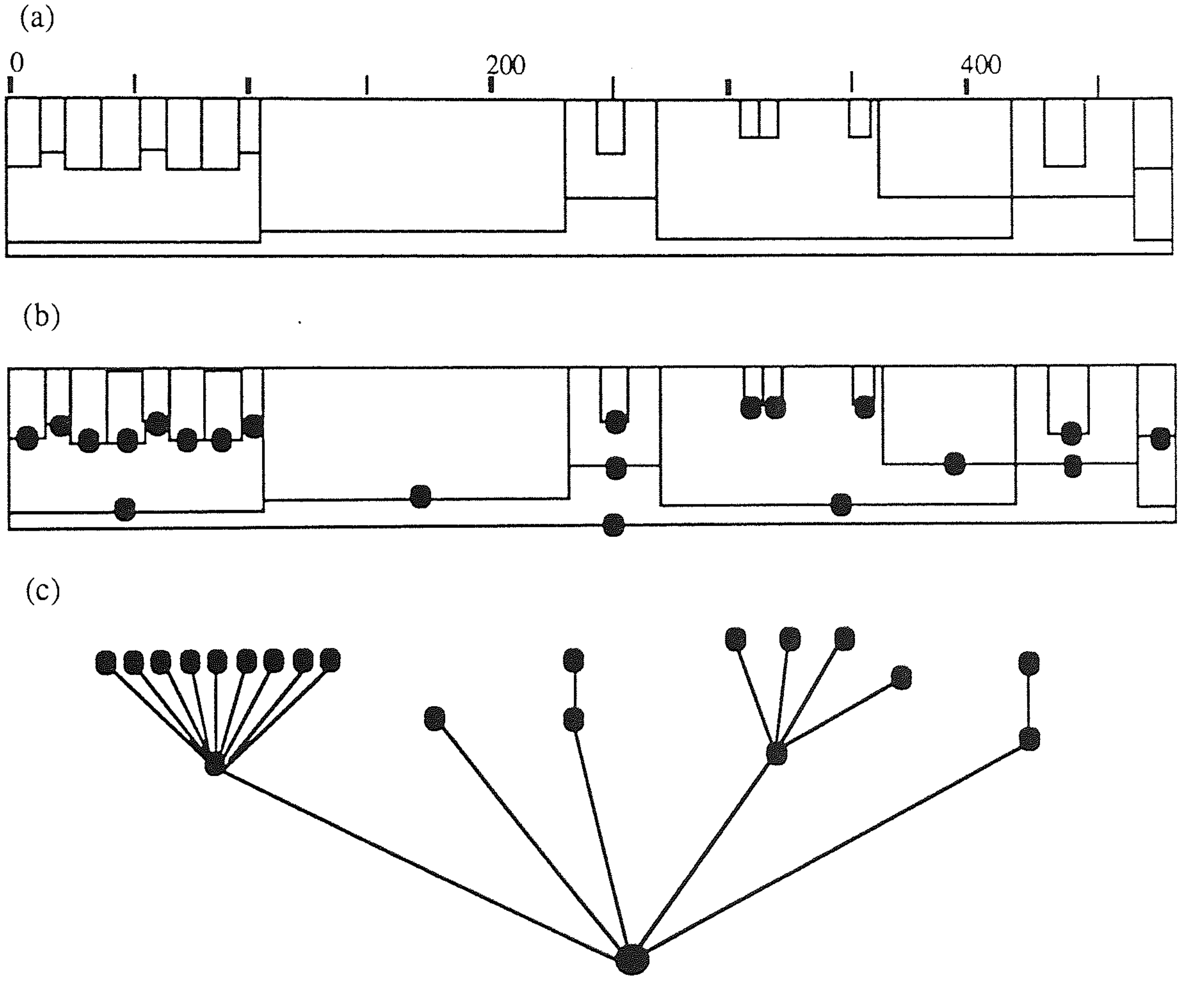


Figure 5.11 Face rectangles and face markers in mmspace, with associated face tree

The nested arrangement of rectangles in mmspace is analogous to a tree structure: a tree node can be identified with each rectangle, and the nested structure will be reflected in the hierarchy of nodes in the tree. Figure 5.11 illustrates the correspondence for a set of rectangles in mmspace. Figure 5.11(a) shows the rectangles corresponding to faces in figure 5.10(a). This figure differs from 5.10(c):

some rectangles in figure 5.10(c) have been divided because the stable zones they represented consisted of two or more adjacent faces.

In figure 5.11(b) a node has been associated with each face rectangle by marking its lower edge, i.e. the face marker. The root node of the tree is different: it represents the entire contour and has been marked on the lower edge of the mmspace rectangle. Figure 5.11(c) shows the corresponding tree; the complete structure may be termed a 'face tree', whether realised as a set of lines in mmspace or as a linked data structure.

It should be noted that the construction of a face tree assumes that the overlap between any pair of faces must be either zero or total: if any point on the contour lies in two faces (excluding endpoints shared between two faces), then one of these faces must be wholly contained within the other. Exceptions to this situation are not common, and so a face tree can be used to provide a useful description of the significant linear structure on the contour. Since linear structure, albeit insignificant, will eventually be found if resolution is made fine enough this technique could provide a hierarchical description of the structure of the complete contour. The face tree would then be analogous to Witkin's interval tree describing the features in a signal, which can be presented in the form of a tessellation of scale-space [5.2.2.1].

5.3.3.1 Choice of arc lengths

One important practical problem in creating the intermediate representation is choosing the set of arc lengths to use. This entails the setting of a lower length limit, an upper limit, and the rule for incrementing the lengths between these extremes.

The shortest arc length used will reflect the highest resolution required. An arc length of 2 is the smallest value which can span a corner; an arc length of 1 can give no useful information. In practice the value 2 was used as the shortest arc length. Arc lengths shorter than the length of the shortest perfectly straight side on the contour provide no additional useful information about the structure, and so the lower limit could be adjusted upwards in some situations, such as the analysis of polygonal shapes, as when the input data has been derived from a polygonal approximation of a contour.

The upper limit used is half the total perimeter (in practice 0.53 of the contour length was used). In the extreme case of an elongated rectangular contour such a value is required for an unambiguous definition of the shape. One possible way (not, however, implemented) of avoiding the use of unnecessarily long arcs is to note the minimum of the mmspace values being generated for successive lengths. If this width divided by half the perimeter (which is approximately the longest possible face length) exceeds a threshold set for the aspect ratio of significant strips, then there is no need to proceed with calculations for longer arcs since they could not serve to find any more faces on the contour. This follows from the general rule that the growth of strip width is non-decreasing with growth of arc length.

The way in which the arc length is incremented is crucially important, since this factor will influence the realisation of stable structure in mmspace. Faces are characterised as contour sections where the width of covering strips is static or grows relatively slowly as the length of the strips increases. Too rapid growth in arc lengths could fail to provide an opportunity for the relative stability in strip width to be detected, and so the face would be overlooked. An unnecessarily slow progression of arc lengths, such as incrementing one chainlet at a time, is also unsatisfactory because each unnecessary arc length used means an unnecessary set of computations in generating the intermediate representations.

A successive doubling of arc lengths was found to be generally successful, and a consideration of corner detection [5.3.3.4] also gave support for choosing a doubling scheme. However, one other factor was taken into consideration which modified this choice: in order for a face to be detectable as a stable region in mmspace at least two successive arc lengths must produce similar minimum strip width values. This is important when linear structure is present at more than one resolution, or when a noisy face contains a relatively smooth section. In order for both levels of structure to be detectable the ratio of their lengths is limited to that of two arc lengths two resolution steps apart. Thus if the arc lengths were successively doubling this ratio would be 1:4, and if a subface were longer than one quarter of its containing face then one of the faces might not be detected.

In order to reduce this ratio the progression of arc lengths at first used two interleaved doubling sequences: 2, 3, 4, 6, 8, 12, 16 ...a, b, 2a, 2b... This reduced the perceptible length ratio to 1:2. This method of interleaving two separate sequences was later modified to provide a scheme which could be more easily generalised, and also gave a smoother increase in arc lengths. A single scaling factor r was introduced to control the progression. Starting the arc length sequence at 2, the recurrence relation between successive arc lengths was a_{n+1} = ceil(a_n.r), where a_n is the nth arc length and ceil() is the ceiling function which returns the smallest integer greater than or equal to its argument.

Using the value r=2 provides the simple doubling sequence of arc lengths; using $r=\sqrt{2}$ gives an approximation to the interleaved doubling sequence. The value r=1.4 was used as an approximation to $\sqrt{2}$, and so the actual sequence of arc lengths used was: 2, 3, 5, 7, 10, 14, 20 ... This arc length progression may be referred to as the slower progression; the doubling sequence may be referred to as the faster progression.

5.3.3.2 Finding stable zones in mmspace

The central problem in analysing mmspace is identifying the stable zones, i.e. the regions of near constant value, elongated in the p-axis direction, which result from two or more successive arc lengths giving roughly similar strip widths over some sector of the contour. One problem in finding such zones is quantifying what is meant by 'roughly similar' width values. It seems reasonable that the difference between two strip width values should be judged in relation to the widths involved, and could therefore be expressed as a percentage of the smaller width. Thus, for example, a given absolute width difference between two narrow, well defined faces would be much less significant if found between two very irregular faces.

One possible first step in the analysis is to differentiate the mmspace values, replacing the strip width values by the change found from one arc length to the next. This, however, did not in practice lead to a more easily interpreted array of values, and also lost the underlying mmspace values. Expressing the width changes as a fraction of the strip widths was similarly unhelpful.

The method suggested by Phillips and Rosenfeld for finding appropriate arc lengths was also considered. This involved dividing the mmspace values by the arc length values used to generate them, and then examining the resultant array for local minima. Although this technique may be useful for suggesting arc lengths of interest it was not found to be suitable for locating stable zones since it produced many line segments which were not true zone markers. This was a result of the fact that any relative increase in strip width which did not exceed the relative increase in arc length would give a decrease in the normalised value (width / arc length). Even if this defect could be cured another problem remained: this method did not directly use the constraint that a valid stable zone is a region of roughly similar value, so locally minimal 'troughs' among the normalised values could abut, even if they represented different zones. This would require a test to be made along each trough, looking for significant steps in the underlying mmspace values.

The approach finally devised for detecting stable zones was based on multiple thresholding, or 'density slicing', of mmspace. An ordered set of thresholds $\{T_0 \ldots T_N\}$ was used to change the relatively smooth surface of mmspace into a 'blocky' approximation. If w denotes a value in mmspace, then $T_i < w \leq T_{i+1}$ for some pair of adjacent threshold values. In the density slicing process the value w would be reset to T_{i+1}. Zero is used as the lowest threshold, T_0, and the maximum threshold, T_N, is at least as great as the maximum value present in the array. Only if w equals zero does w remain zero following the density slicing. However, in order to avoid accuracy problems with values near zero, a small threshold had been used in the production of mmspace, re-setting any value below 0.2 to exactly zero. Thus the overall effect of the density slicing is that (near-)zero mmspace values are set to zero and any other value strictly exceeding a threshold T_i is rounded up to the next threshold value T_{i+1}.

A modification of this density slicing scheme, which dispensed with the category for faces of zero width, was also experimented with. The use of this modification is discussed below [5.3.3.5].

In practice it was found to be useful to reset the mmspace values to the index of the threshold in the ordered threshold set, and not to the actual threshold value. This process transforms the range of real values found in mmspace into integers in the range 0 to N. It then becomes relatively convenient to examine the connected regions of constant value in the discretised mmspace, to see if they have the characteristics of stable regions. If the set of thresholds is carefully chosen the density slicing process could result in all values within a stable zone being reset to the same threshold (index) value, so giving a perfectly flat region sharing the geometric properties of the original zone but in a more easily analysed form.

For each integer n in the range [0, N−1] a pass is made over the density sliced version of mmspace, taking each row of constant a–coordinate in turn. Travelling along these rows, points are sought where the value n is produced both for this arc length and the previous, smaller arc length, and where the next, larger arc length produced a higher value. These points are marked in a corresponding array, so building up a picture of potential zone markers for this particular threshold interval. At this stage invalid zone markers can be recognised on the grounds of their length, which ought not to be smaller than the generating arc length, and deleted.

Thus the density slicing approach allows stable zones to be picked out one by one, and retains the underlying strip width value associated with each zone. Implementation of the density slicing and the representation of zone and face information are discussed in detail below [5.3.6.1].

5.3.3.3 Choice of thresholds

The above scheme for identifying zones in the density sliced mmspace array appears to rely on a good choice of thresholds. An ordered set of thresholds defines a set of intervals, or 'bins', between adjacent thresholds. It can be seen that the problem is to choose the thresholds in such a way that the values in any one stable zone in mmspace all fall into the same bin, and so all become rounded up to the same ceiling value.

The observation that similarities in strip widths should be judged in relation to the widths involved translates into the constraint on the thresholds that the interval between adjacent thresholds should be proportional to the lower threshold value. This condition implies that the threshold values will grow in a geometric progression, and the bins will become increasingly large. If the proportion is set at 50%, for example, and given a first nonzero threshold value of 3, this rule would generate the sequence of thresholds: 0, 3, 4.5, 6.75, 10.13, 15.19,... a, 1.5a, 1.5^2a... Setting the proportion at 100%, and using the same first non-zero threshold value, generates the sequence of thresholds: 0, 3, 6, 12,... a, 2a, 4a...

Values of both 50% and 100% were tried to define the bin widths as a fraction of their floor values. Both were found to be generally successful for contours with relatively little noise. For noisier contours, however, the 100% factor was found to give better results.

An important value which must also be selected is the first nonzero threshold value. This value could not be made too small, since even an extremely straight line may require a covering strip width of 2: a value in the range 2 to 4 was considered appropriate.

The requirement for good thresholds cannot in the general case be satisfied by a single set of values. Even within one contour, two stable zones may require corresponding bin limits which overlap, say from 3 to 4.5 and from 4 to 6. It was therefore found necessary in using the density slicing approach to employ two or more sets of thresholds, and introduce additional processing to deal with any conflicting or redundant interpretations which may subsequently arise. (The additional processing requirements are discussed in section 5.3.4.)

Given that several sets of thresholds will be required, one possible approach which merited investigation was to try to identify the important values in mmspace, i.e. those which give the heights of likely stable zones, and then design one of the sets of thresholds in order to deal optimally with each likely zone. Information about the heights of stable zones could be gained by finding the mmspace values corresponding to small derivative values of strip width as a function of arc length. Such adaptive processing was, however, found to be unnecessary. Satisfactory results were achieved by a more straightforward method.

The approach adopted was to use three sets of thresholds which produced an overlapping sequence of bins. When the 100% value was used to define the bin widths the following family of threshold sets was used:

TS1 = (0, 2, 4, 8, 16,...)

TS2 = (0, 2.67, 5.33, 10.67, ...)

TS3 = (0, 3.33, 6.67, 13.33, ...)

These sequences are such that any bin of non-zero floor value is overlapped to approximately two thirds of its extent by two other bins. This overlapping arrangement means that the clusters of mmspace values associated with a stable zone are likely to fall together into at least one of the bins. Some trials were also made using families of 5 threshold sequences, but no clear improvement was found.

5.3.3.4 Corner detection

Finding if a stable zone represents one or more faces is achieved by checking for the presence of significant corners within the stable zone. This brings two immediate problems: choosing the right resolution at which to look for corners, and evaluating the importance of any corners found. For the analysis of ideal polygonal contours a wide range of resolutions (arc lengths) could be used for successful corner detection. When real, noisy contours are being analysed the problem of choosing the appropriate resolution has been seen to be not straightforward: this was the problem considered by Davis, Fischler and Bolles, Phillips and Rosenfeld. The new dual-component intermediate representation of side and corner structure proposed here can be used to provide a solution.

As an example, it may be supposed a zone marker in mmspace extends from p1 to p2, has associated arc length c, and associated strip width value w. Then any point on the contour between p1 and p2 lies in an arc spanned by c chainlets which can be covered by a rectangular strip no wider than w. This knowledge of the straightness of the contour segment and of the arc length used in making the measurement can be used both to select an appropriate arc length in the corner measurements and to set a threshold on what is a significant corner value.

An appropriate arc length to use in the corner analysis is 2c, and the row of cpspace on which the search window should be set up can therefore be defined. To see why 2c is a suitable value it may be recalled that the value held in the cpspace array for a contour point p and arclength 2c is the maximum perpendicular distance from p to the chord of an arc of length 2c which contains p. If p is at a significant corner within a stable zone then, following the reasoning given for figure 5.YY above, its domain must be approximately symmetric. The limbs on either side of the corner are therefore approximately of length c. Figure 5.12 illustrates this:

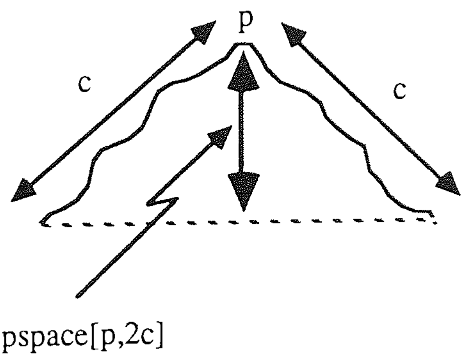


Figure 5.12 A corner within a 'stable zone'

The two limbs of the corner, having lengths approximately c, can each be covered by a strip of width approximately w. Thus w can be used as a threshold on the significance of corner values within the search window. As in the ideal polygon case discussed above, the cpspace values will be examined, for arclength values of 2c, for p coordinates within a subwindow of the range (p1,p2). The full range need not be examined since the immediate neighbourhoods of the zone endpoints (themselves breakpoints) will not contain significant corners for this resolution. When the maximum value in the window is found it is compared with the threshold w. If it exceeds the threshold it is accepted as a corner, a small neighbourhood of this point is excluded from the search window, and the search continues for the next largest value. The process is repeated, finding maxima and reducing the search zone. The process terminates when the maximum value found in the window is below the threshold level, or the search window has been reduced to extinction. The implementation is described in greater detail below [5.3.7.1].

Thus the location of stable zones in mmspace can be used to simplify the analysis of cpspace. Establishing a stable zone marker in mmspace has the effect of defining corners at the zone limits. As a result, corners with asymmetric domains will be found at this first stage, with no reference being made to cpspace data. Corners with roughly symmetric domains must lie within stable zones and this extra constraint has been used to place the search window on the cpspace array, evaluate the size of the locally maximal values found there, and efficiently focus the search until significant structure is extracted. This approach has also eliminated the need for smoothing or processing of any kind on the cpspace values, apart from local maxima location.

5.3.3.5 Illustrative examples: finding possible faces

The analysis of some test profiles by an implementation of the multiresolution algorithm will now be considered. In this section the recognition of stable zones and faces will be demonstrated using both the slower (r=1.4) and faster (r=2) arc length progressions. The results of different density slicing strategies are also shown. The processes by which the sets of hypothesised faces which are the results of this stage of processing are winnowed and refined to produce a useful description of the linear structure on the contours will be considered below [5.3.4].

The first test profile considered here is synthetic; the second and third are real crystal profile images. Section 5.3.4.5 will show the results of later stages of processing on these examples. Section 5.4.2 will show the results of the complete analysis for these examples and for a further two synthetic images and eight real crystal profile examples.

Synthetic data example

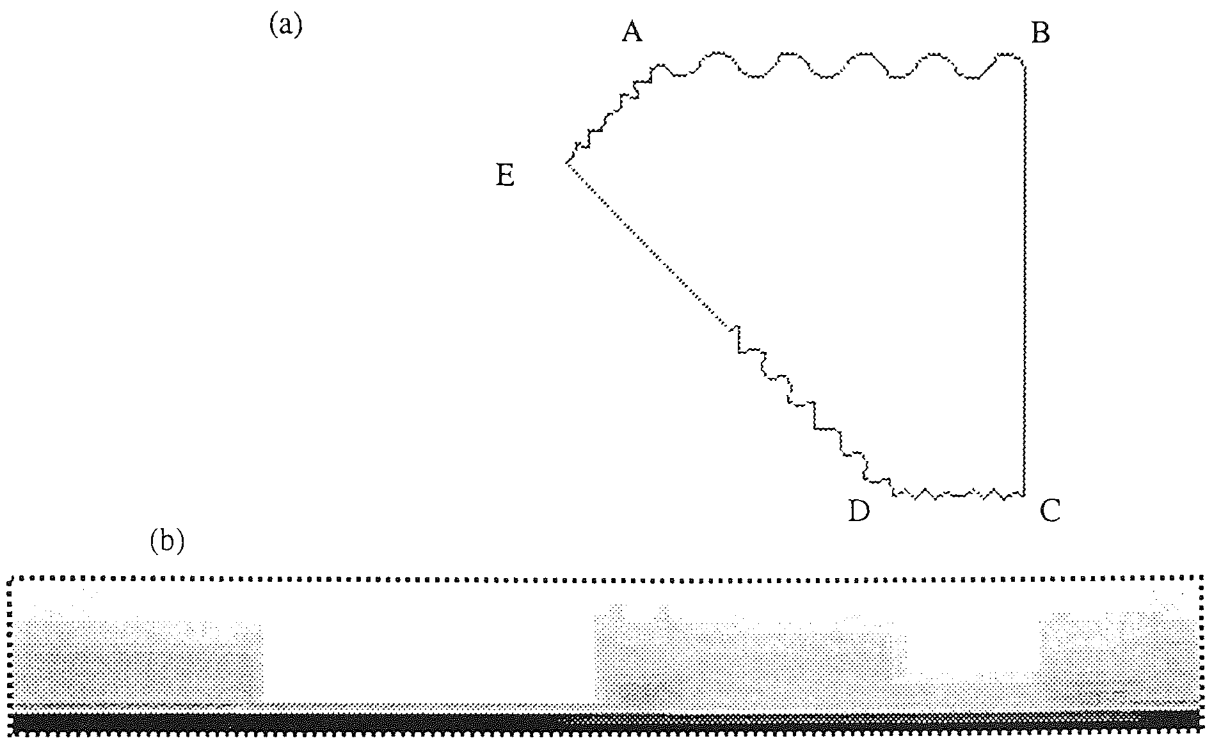


Figure 5.13 'Pentagon' test contour and its mmspace representation

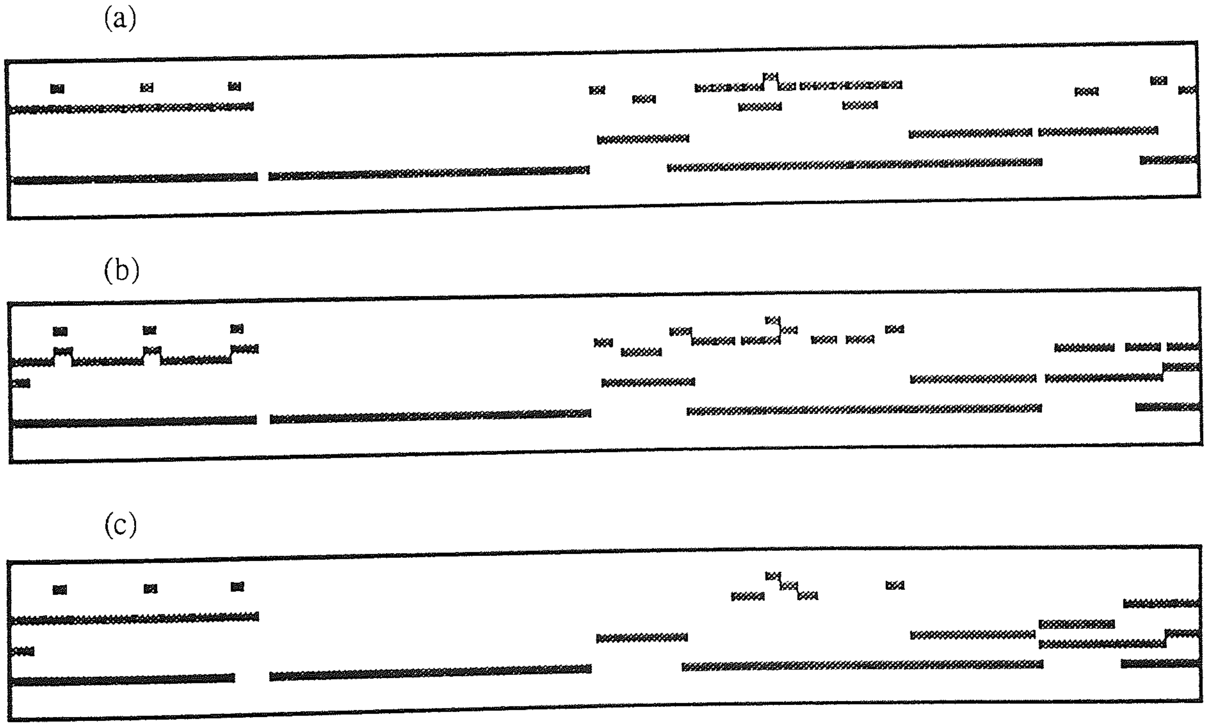


Figure 5.14 Face markers in mmspace found via three sets of thresholds, using the slower arc length progression

Figure 5.13(a) is a synthetic test contour which shows linear or near-linear structure at more than one resolution; this figure has already appeared as 5.10(a). The 'vertices' have been labelled for later reference. Figure 5.13(b) repeats 5.10(b): this shows mmspace for this contour, with higher values indicated by the darker grey tones. The slower progression of arc lengths was used to create mmspace, i.e. a sequence of 15 arc lengths ranging in span from 2 to 307 chainlets.

Figures 5.14(a), (b) and (c) respectively show the face markers found in mmspace after density slicing with the threshold sets TS1, TS2 and TS3, described above [5.3.3.4]. Each face marker is terminated by darker dots so that the extent of adjacent markers can be properly identified. This reveals how zone markers have been split into contiguous face markers.

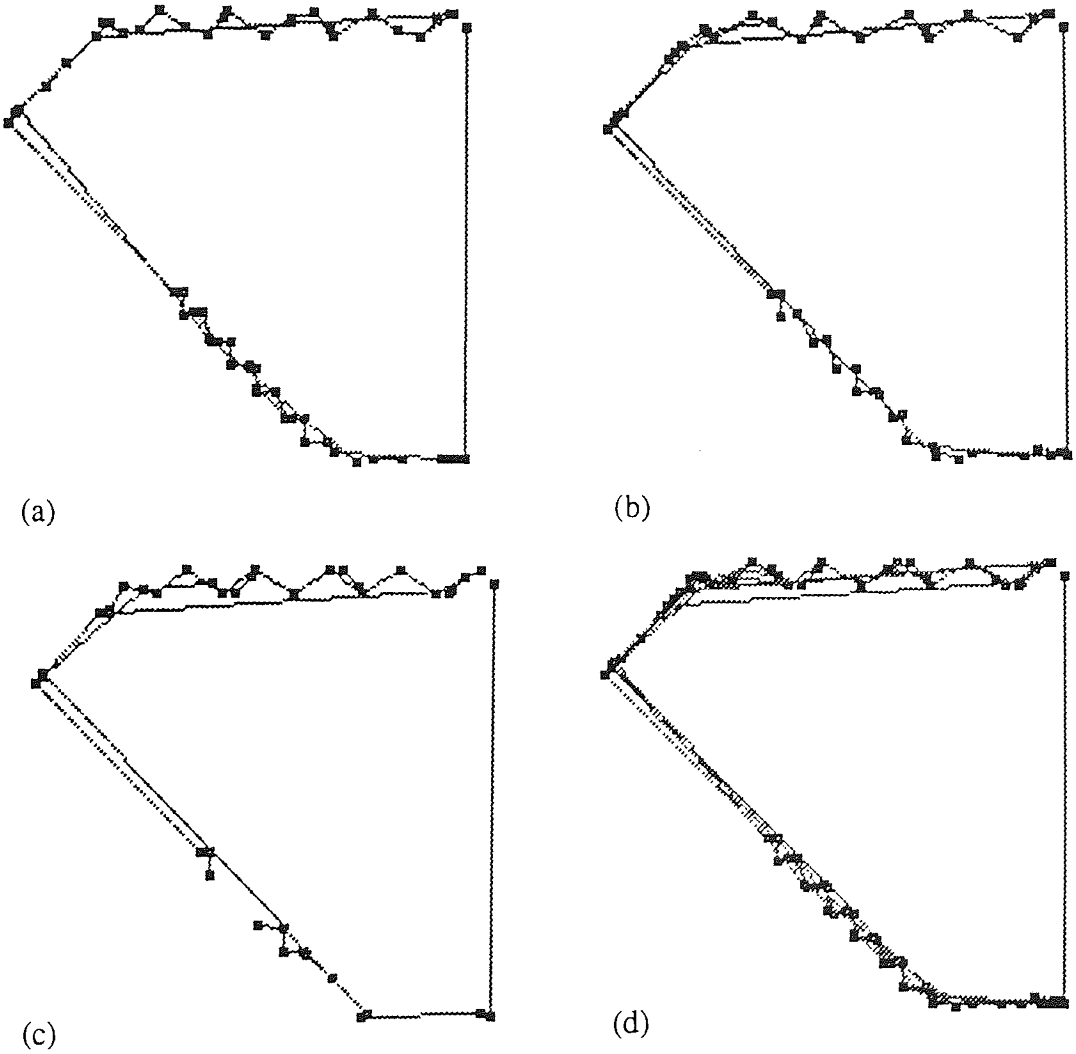


Figure 5.15 Faces corresponding to the markers of 5.14, and their superimposition

Figures 5.15(a), (b) and (c) show the sets of (possible) faces corresponding to the face markers of figure 5.14. The displays of figure 5.14 and 5.15 have been pruned of faces of true (Euclidean) length less than the equivalent of 6 chainlets, or aspect ratios greater than 0.33. The latter limit is more generous than would probably be used in the practical analysis of crystal profiles, but has been used here in order to give a better indication of the range of structures which it is possible to discern. Figure 5.15(d) shows a superimposition of all the suggested faces found for the three density slicing passes. This figure therefore represents the product of the first stage of the analysis: the set of faces which must be reduced by further processing to yield the set of best estimates, with no contradictions or redundancies in the representation.

It should be noted that the extent of possible faces on the contour is marked by drawing a chord between the estimated endpoints; the resulting line segments do not necessarily represent optimal approximations to the faces. For very noisy faces some additional processing may be required to find a more representative, better fitting line segment. This will be considered below [5.5].

The face markers of figure 5.14 allow the structures seen in figure 5.15 to be referred back to mmspace (figure 5.13(b)), and also elucidate the structure where many faces overlap. Figures 5.15(b) and (c) show slight overestimates of the extent of face DE. This is because, for longer arc lengths, the mmspace values in the stable rectangle shelve upwards towards the neighbouring face; the rectangle does not terminate abruptly. Thus any threshold value above the 'floor' of the rectangle will also encompass part of the slopes, producing a corresponding face marker which encroaches into the neighbouring face. The reason for this shelving is that in the analysis of points on face CD lying near D, when arc lengths greater than |CD| are used, the minimum covering strip width will be delivered by strips which lie along face DE and not CD. The bevelled termination of stable rectangles in mmspace is considered again below [5.3.4.3].

In figure 5.15(a) face DE has been found as two faces. Referring to the corresponding face markers of figure 5.14(a) shows that one zone was found for the face, but the zone marker has been split into two face markers. However, since the other two threshold sets led to a correct identification of the whole face, the subsequent processing stage which combines the results of the tree passes will provide the correct final interpretation (figure 5.25 below shows the results of the later processing stages).

In a similar fashion, although the third threshold set found face DE correctly, it has proved inappropriate for analysing the irregular portion of that face: the irregular segment has been analysed as portions which fail to meet the thresholds on length and aspect ratio, and which have therefore been deleted. Since the other two sets have provided good analyses of this segment, the next processing stage should be able to find the correct interpretation (this will be shown to be the case [5.3.4.5]).

The analyses can also be seen to pick out relatively smooth subparts of irregular portions. Both faces EA and CD show slightly smoother central zones; these have been identified by threshold set TS1. TS2 yields a slightly different description of the structure of face EA. Choosing the best interpretation from the alternatives offered is left to the next stage of processing.

The density slicing passes used in these analyses all maintained a special category for perfectly straight faces, i.e. faces of zero strip width. Consequently, the two long, perfectly straight contour segments have been found correctly in all three cases, with no overestimates of their extent. Rejecting the use of the zero-width 'bin' in the density slicing would be expected to lead to overestimates of the extent of faces which were perfectly straight: the effects may be seen in the processing of the second real crystal profile below.

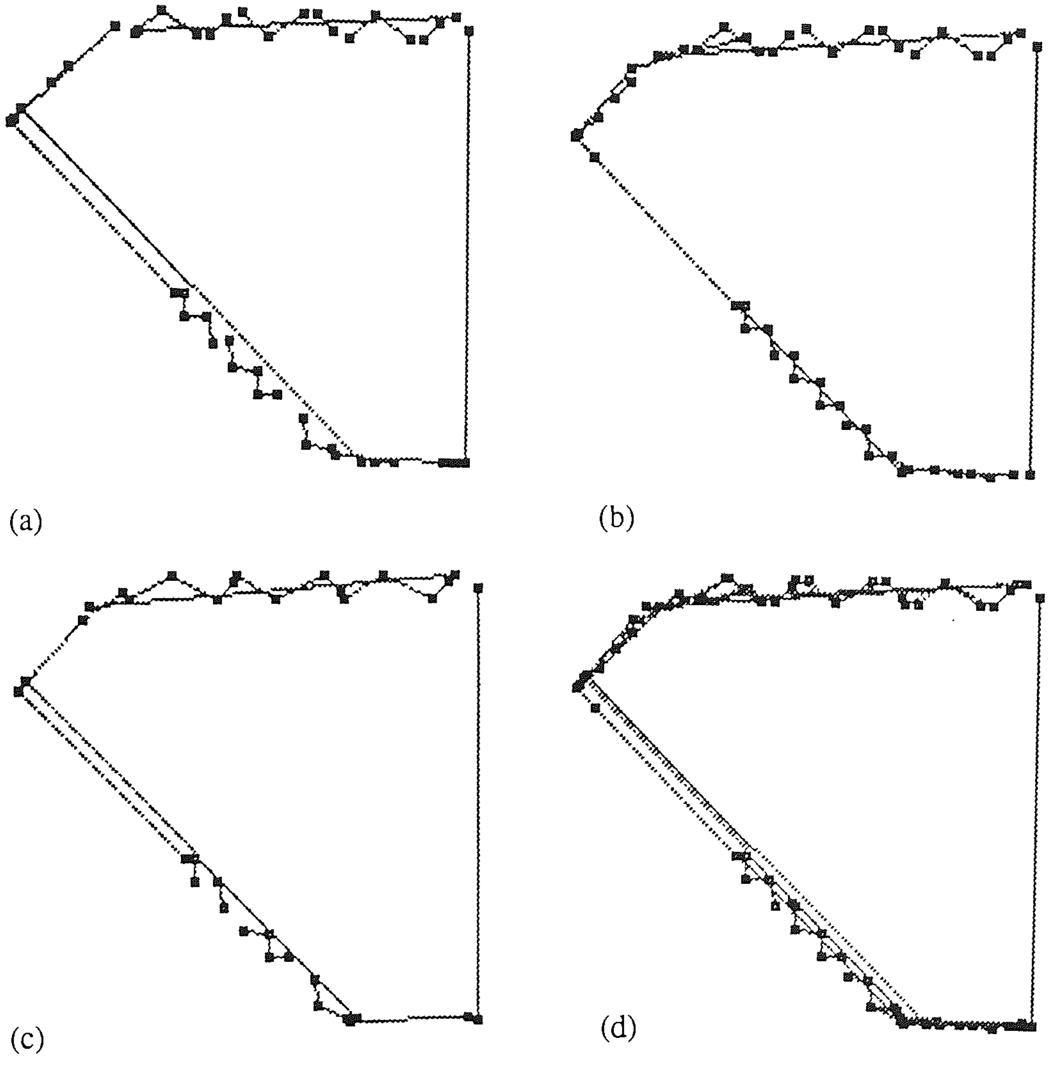


Figure 5.16 Faces found via three threshold sets and their superimposition, using the faster arc length progression

Figure 5.16 presents the results of the same face-finding procedure applied to the contour, but using the faster, doubling arc length progression in the creation of the intermediate representation: only eight arc lengths were used, ranging in span from 2 to 256 chainlets. In all other respects the processing is identical to that which produced the results of figure 5.15. Figures 5.16(a), (b) and (c) show, respectively, the results of processing using threshold sets TS1, TS2 and TS3; figure 5.16(d) shows the superimposition of the three sets of faces found.

As discussed above [5.3.3.1], in using the more rapid progression of arc lengths one would expect the loss of less stable faces, and yet in this case the change has made relatively little difference - even some of the substructures of faces EA and CD have been picked up. The face DE has been correctly identified on each pass, although the full extent of the face has not been found in the case of the second threshold set (figure 5.16(b)). Referring to mmspace (not shown), a very slight rise could be seen in the stable rectangle of values representing the face, at arc length 128. One of the thresholds used has unfortunately separated this higher-valued patch from the rest of the rectangle. This has resulted in the face marker being shorter than required. The unevenness in the stable rectangle reflects variations in the irregular portion of face DE. This shows that the minimum covering strip width value can be affected by contour features at the other end of the arc used, here 128 chainlets distant. The other two threshold sets have found all values within the stable rectangle to lie in one inter-threshold bin, and the extent of the face was correctly found.

Real data example: topaz crystal

Figure 5.17(a) shows a real crystal profile image, that of a topaz crystal. The image is derived from a published photomicrograph [MacKenzie & Guilford 1980, example 17]. This figure was created by using a video camera to produce a 256 grey level digital image of the photograph, then density slicing the image to isolate the desired crystal. The resulting binary image was then cropped to a 150 X 150 array, and the perimeter of the crystal was extracted using a version of William's [1978] approximation algorithm with the error parameter set to zero. The boundary was not smoothed in any way.

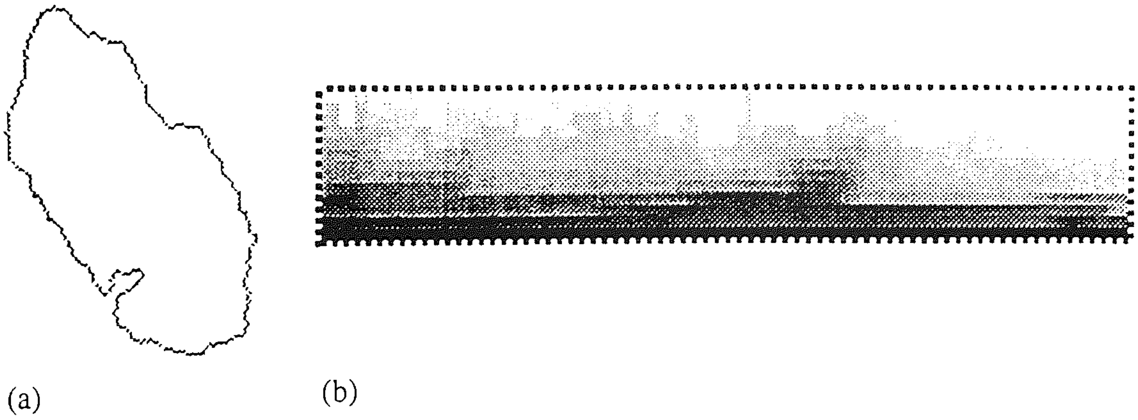


Figure 5.17 Topaz crystal profile and its mmspace representation

Figure 5.17(b) shows the mmspace representation corresponding to figure 5.17(a). The origin of the chaincode is at the topmost point on the contour. The contour is encoded clockwise; there are 286 chainlets in the resulting chaincode. The slower progression of arc lengths was used to create mmspace, i.e. a sequence of 13 arcs, ranging in span from 2 to 156 chainlets. The parameters of the analysis were identical to those used for the synthetic test contour, apart from the aspect ratio limit set on acceptable faces. The aspect ratio limit used was 0.25, which was the value used in the analysis of all real data examples.

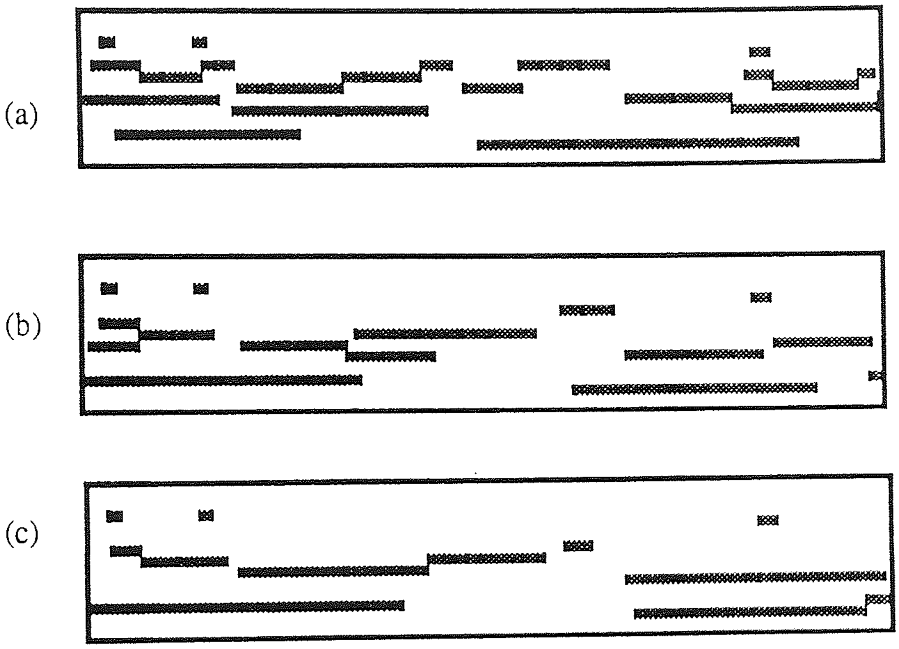


Figure 5.18 Face markers in mmspace found via three sets of thresholds, using the slower arc length progression

Figures 5.18(a), (b) and (c) show the face markers found using the three threshold sets, respectively TS1, TS2 and TS3, minus the markers for faces of lengths less than 6 or of aspect ratios greater than 0.25. Figures 5.19(a), (b) and (c) show the corresponding sets of faces.

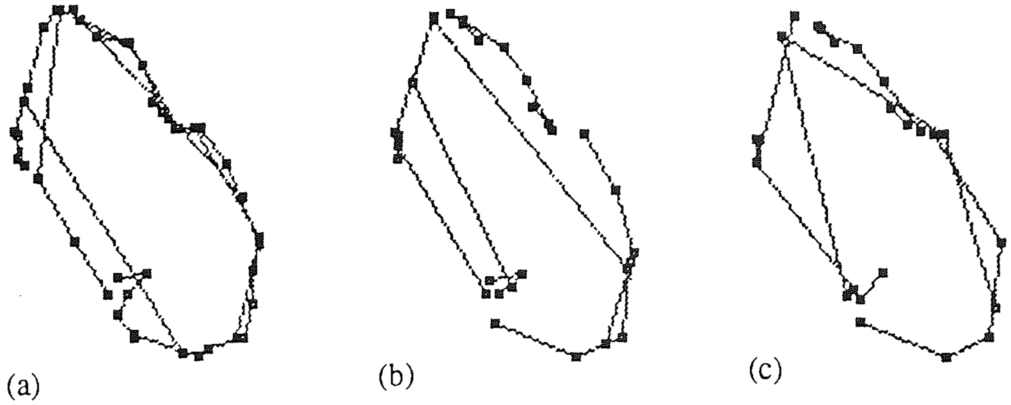


Figure 5.19 Faces corresponding to face markers of figure 5.26

It can be seen from figure 5.19 that no single threshold set produces a complete set of the required faces. Using TS1, the structure of the right side of the figure appears to be described quite well, but the analysis of the left side either fails to find the better faces, or valid faces appear to have been split unnecessarily. Using TS2, the algorithm has identified the faces on the left side quite well, but has produced a poor analysis of the structure on the right side. Using TS3, the algorithm has made at least six good estimates of valid faces, along with suggestions for less probable faces.

Taken together, however, the three analyses appear to have found all of what one would reasonably judge to be the preferred faces, together with some useful detail of possible subfaces. The suggestions also include some poorer estimates of faces, but these, it would be expected, can be removed in the later processing stages. Since the subsequent processing steps devised [5.3.4] can only delete or adjust slightly the faces found in this first stage of processing, and cannot make entirely new suggestions for possible faces, it is better that the earlier processing should be biased towards making too many suggestions of faces rather than too few.

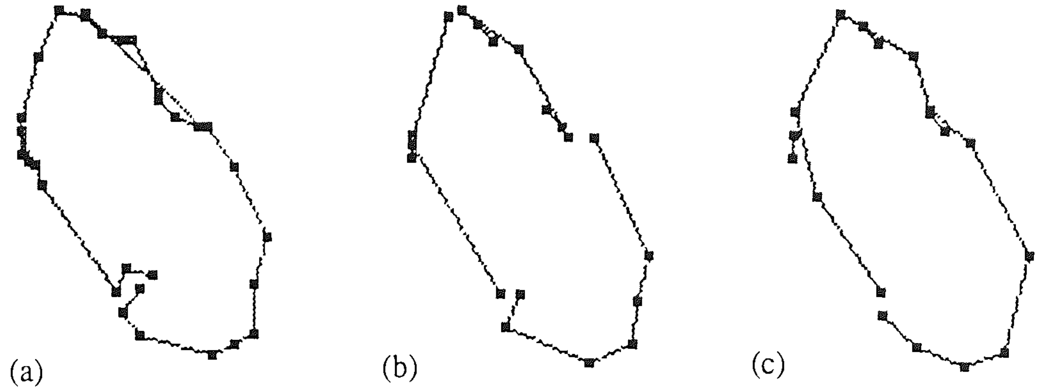


Figure 5.20 Faces found via three threshold sets, faster arc length progression used

Figures 5.20(a), (b) and (c) show the results of repeating the multiscale analysis of the topaz crystal profile, using the faster, doubling sequence of arc lengths. Eight arc lengths were used in the creation of the intermediate representation, ranging in span from 2 to 256. The processing parameters are otherwise identical to those used in the production of figures 5.19(a), (b) and (c) respectively.

The major difference between the results found using the slower progression and those found with the faster progression is that fewer faces of high aspect ratio are found when the faster progression is used. The oblique face at the top right side of figure 5.20(a) is the 'widest' face found. Such a loss of the less 'stable' faces was anticipated [5.3.3.1].

For this image at least, considering the results of the three analyses together, it seems that using the faster progression has not led to the loss of the most important faces. The results of the later processing stages confirm this [5.3.4.5; 5.4.3].

Real data example: lamprophyllite crystal profile

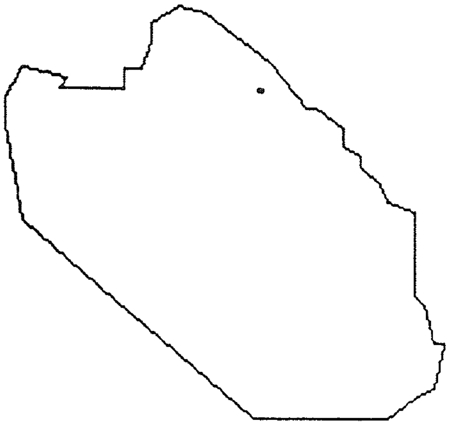


Figure 5.21 Lamprophyllite crystal profile image

Figure 5.21 shows the profile of a lamprophyllite crystal. This image was derived from a published photomicrograph [MacKenzie & Guilford 1980, example 53] using the digitisation facility of an AutoCAD package. Digitisation produced a polygonal approximation to the contour in the form of a list of vectors. The polygon was scaled to fit within a 180 x 180 square, and the vector image was rasterised to create the final digital image.

The hand digitisation of a contour is liable to introduce error since the approximation made may be more irregular in parts than is appropriate, or over-simplified by using digitisation steps which are too large. However, a useful analysis scheme should be able to cope with such inexact data, and figure 5.21 should therefore be an acceptable image for the multiresolution analysis algorithm. The algorithm is designed to accept the chain-coded representation of a contour and the method of generation should be irrelevant.

Figure 5.22(a) shows a superposition of all the faces found by the three threshold sets, excluding faces of length less than 6 chainlets or aspect ratio greater than 0.25. The slower arc length progression was used in the creation of the intermediate representation: a sequence of 15 arcs lengths ranging from 2 to 307. As in the processing of the previous examples the density slicing procedure used a special category for faces of zero strip width.

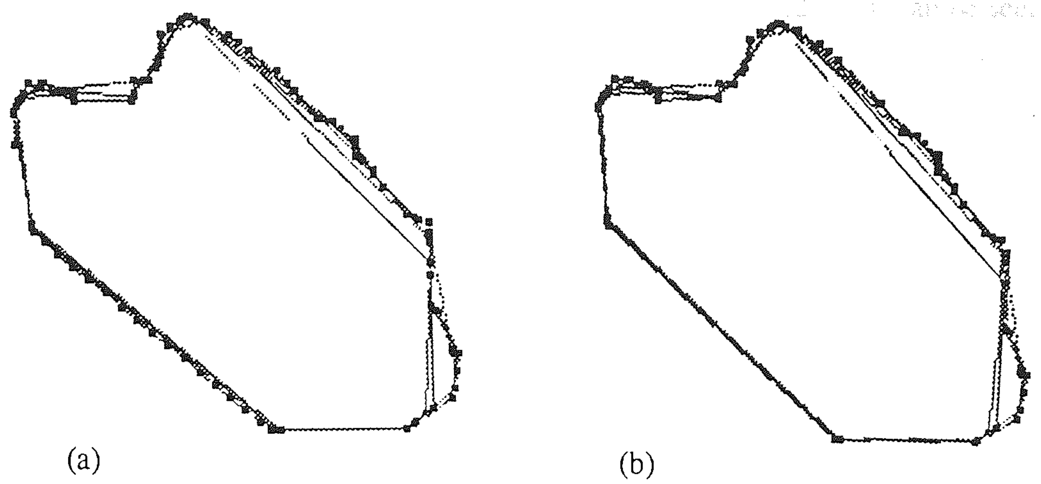


Figure 5.22 Faces found from three density slicing passes, with and without special category for perfectly straight faces

The algorithm would appear to have produced a good description of the linear structure on the contour. The straight segments of the contour have been found. A long, slightly irregular upper face has been found which, lying approximately parallel to the lower face, appears to be a valid and important face. Another irregular face has been hypothesised on the right side of the figure, which again seems to be valid in view of the crystal symmetry, and probably preferable to its straighter subfaces, although the suggested face is slightly too long. However, more than a dozen small subfaces have been found on the long straight lower face. These subfaces represent the perfectly straight segments by which the oblique line is realised on a discrete pixel grid.

It is not acceptable to remove such subfaces by simply increasing the face length threshold, since valid faces elsewhere on the contour could be lost. Another strategy considered was to suppress the generation of these subfaces by modifying the density slicing procedure and removing the 'bins' reserved for perfectly straight faces.

The effect of this modification on stable zones which represent only isolated single straight faces may be considered first. With no category in the density slicing allotted for faces of zero width these zones will fall into the first inter-threshold bin. Using the threshold sets described above [5.3.3.3], the three bins into which the zones will fall have limits [0, 2], [0, 2.67] and [0, 3.33]. Since the upper limits of these bin intervals are greater than zero, the zones found will almost certainly represent the straight face plus some part of the neighbouring contour. If another straight contour segment lies parallel to the first and the two are separated by only one chain code chainlet, then the stable zone should extend to represent both faces. In this way the series of subfaces created by the digitisation of a long oblique straight line should be found as one stable zone.

Figure 5.22(b) shows the effect of analysing the contour of figure A using the density slicing algorithm minus the bins for perfectly straight faces, and using the same face length and aspect ratio limits as before (i.e. 6 and 0.25). It can be seen that this modified density slicing scheme has successfully suppressed the generation of the short, perfectly straight facets within the long oblique lower face. The negative aspect of the modified scheme should also be noted, however. The short, perfectly straight vertical face on the right of the figure has been found, but small portions of the adjacent contour have been included in the estimate, so the face endpoints define an oblique face of non-zero width. Similarly, the short, perfectly straight horizontal face within the embayment on the top left of the figure has been found, but again incorporating parts of the bracketing contour, and the face endpoints which result define an oblique face of non-zero width.

Thus it can be seen that suppressing the 'step faces' produced in the digitisation of oblique lines, by dispensing with the category for perfectly straight faces in the density slicing process, brings the unwanted side effect of loss of accuracy in the recognition of valid straight faces. An alternative strategy for eliminating these unwanted faces was therefore pursued: not to attempt to suppress their generation, but to use additional steps in the later stages of processing to recognise and remove them. Such processing is considered below [5.3.4.4].

5.3.4 Combining structural interpretations

To recapitulate section 5.3.3, stable zones in mmspace result when two or more consecutive arc lengths give little increase in strip width, measured as a proportion of that width. By applying a set of thresholds to mmspace, stable zones become transformed into perfectly flat regions. Three overlapping sets of thresholds are used to give good coverage of possible stable zones. Each threshold set produces its own estimates of the stable zones. These three sets of zones may then be examined in turn to find the faces in the zones. This involves corner detection within each zone.

This analysis will therefore produce three sets of likely or possible faces, found via the three threshold sequences. These three interpretations must now be combined, eliminating redundant representations and resolving any conflicts. The result of the combination should be a description of the significant linear structure in the contour. If such structure exists at more than one level of resolution, then the hierarchy should be preserved, ideally to yield the face-tree representation described above [5.3.3].

The description of the algorithm which was devised for combining sets of face estimates assumes 'face record' data structures. As was described above [5.3.3], each stable zone can be represented by a line segment in an array whose dimensions match those of mmspace. In the same way, faces can be represented by line segments in an array. Such a representation, however, makes subsequent processing inconvenient and inefficient, since the information describing the face is implicit and must be measured on the array. It is much more convenient to produce explicit descriptions of the faces and store this information in record or list structures. In the actual implementation (described in greater detail below [5.3.8]) linked record structures were used to represent both zones and faces.

It will therefore be assumed that records have been produced to describe each (possible) face. All relevant face attributes are stored in the records: start and end points on the contour, length given as a number of chainlets, the true length (Euclidean distance between endpoints), width (an approximation in the form of the threshold bin ceiling value), and aspect ratio for the covering strip, calculated as the quotient of the width and true length values. The direction of the face may be recorded if required. A field in the record also records the a-coordinate of the face marker. No attempt is made at this stage to model in the records any hierarchical structure between faces.

5.3.4.1 Preliminary filter

The first step towards describing the significant structure is a simple filtering pass over the data, removing very short faces, or faces of high aspect ratio. Although this 'cleaning up' of the data is relatively straightforward there are some alternative strategies which could be considered. When removing short faces true lengths could be used, or lengths measured along the arc. Since the lengths are likely to be similar it is preferable to use the latter measure, because this allows the calculation of true lengths and aspect ratio to be performed only for faces which pass this test. This will remove the unnecessary computation involved in filling in the attribute list for insignificantly small facelets.

Length thresholds (lower limits) may be of two types: absolute or relative. An absolute threshold is the same for all contours; a relative threshold is specified as a proportion of the total perimeter of the contour being analysed. The two thresholds could be combined by taking the maximum. An examination of real images suggested that for the analysis of crystal profiles a relative length threshold was inappropriate since the faces present could be imposed by occluding crystals of different character and whose sizes are independent of the profile being examined.

In this analysis, therefore, only a simple, absolute threshold was used. The value of 6 chainlets was used This value represents 2% of the length of the smallest contour used. It may be appropriate to reduce this threshold further for smaller figures, but no tests have been made.

The aspect ratio threshold (upper limit) is independent of the face length. This is in accordance with Lowe's [1987] practice. However, given a set of faces with identical shape and aspect ratio, the longer ones will appear the more significant, by virtue of their length. The idea of using an aspect ratio threshold which was a function of face length was therefore considered. The decision not to use such a scheme was mitigated by the knowledge that the symmetry analyses which will be applied to the final output could be made to take relative length into account. The value of the absolute threshold eventually decided upon was 0.25, which was found to be reasonably effective in practice.

5.3.4.2 Elimination of conflicts and redundancies

With the pool of possible faces pruned of insignificant members the process of choosing the subset which gives the best description of the contour structure can begin. The scheme finally devised to achieve this used a relatively simple 'greedy' algorithm. The meaning of the epithet will become clear in the following description.

In its simplest the algorithm begins by grouping together all the records of the candidate faces according to their a–coordinate values. For each such group the face records are then arranged in order of decreasing face length. True length (i.e. Euclidean distance between endpoints) is used rather than pseudolength since it is a better reflection of the importance of a face. This produces a list which may be called the candidate list. From this list a list of accepted faces will be constructed: the 'pass list'.

The greedy algorithm proceeds by taking the first face record in the candidate list, which may be termed the leading face, and transferring it to the pass list. First, however, it is compared with all other faces represented in the candidate list. If the leading face does not overlap a candidate face then the candidate face survives unaltered. If the leading face overlaps a candidate face completely then the candidate face record is deleted. If the leading face overlaps a candidate face only partially then the overlapping section is removed from the candidate face, and the face record attributes are updated accordingly with new endpoint information, length, pseudolength and aspect ratio values. If the length or aspect ratio becomes such that the face in this form would not have survived the preliminary 'significance' filter then it is deleted; otherwise it persists in its new truncated state.

When the leading face has been compared with all members of the candidate list and the required deletions and truncations made, the candidate list is re-ordered. This is required to counter the disordering effect of any truncations of the candidate faces. The new leading face is then taken from the head of the candidate list and the process is repeated. The process terminates when the candidate list is entirely consumed. The pass list then holds a set of face records representing a non-overlapping collection of faces for this particular arc length which is the best according to the greedy principle.

The candidate faces may have had widely different strip width values, since common a-coordinate is no guarantee of similar mmspace values. However, the comparison process described above is valid because any *overlapping* faces with the same a-coordinate must have similar widths, since they share the same mmspace value along their overlap. If faces do overlap then the use of a common a-coordinate is sounder than using the faces' associated strip width values, since these are threshold values giving rounded up approximations to the true width, and different threshold sets will give different approximations for the same underlying mmspace values.

This greedy algorithm can be improved by making further use of the stability property of good faces. Instead of building candidate lists for faces representing the same single arc length value only, the arc lengths can be taken in adjacent pairs. This exploits the fact that stable regions in mmspace must, by definition, be supported by two or more adjacent arc length values. If two faces found by adjacent arc lengths overlap, this property implies that they must have similar widths, and must have been found by different sets of thresholds. Since they represent the same zone and should not overlap only the longer is chosen.

Considering the arclengths in adjacent pairs therefore helps to pick out the best face for each stable zone. If two overlapping faces are found at the end of this process then it can be guaranteed that their arc lengths are not adjacent, and so they should represent structure perceptible at different resolutions.

In implementing this version of the greedy algorithm two passes are made over the face records. If, for example, a doubling sequence of arc lengths were used from 2 to 64 chainlets, the first pass would consider faces for arc lengths 2 and 4, 8 and 16, 32 and 64. The second pass would consider the pairs 4 and 8, 16 and 32. This approach of analysing face record in groups defined by the arc length values has computational advantages since it reduces the number of comparisons which need to be made between faces to separate the various levels of significant structure.

It may be remarked that an early analysis scheme attempted to resolve the different levels of structure in a similar way, but considered faces in groups of similar width. Neglect of the a-coordinate information greatly increased the searching required. Further, the width values are only approximations and were not always sufficiently accurate. As has been shown, equal or adjacent a-coordinate values guarantee similar width of overlapping faces, without the need to find out what the width actually is.

Excluding straight faces from the greedy selection

Since it had been observed that removing the bin for perfectly straight faces could occasionally improve the definition of faces found by the first threshold interval, an alternative version of the analysis algorithm was implemented which tried to combine the advantages of the two density slicing schemes. First the density slicing was performed without the bins for straight faces, and these faces were pruned of poor faces and passed on to the greedy selection procedure as before. Secondly a pass was made over the intermediate representation to extract only the perfectly straight faces. This set of faces was pruned of members which did not reach the minimum length threshold. The list of records for the surviving straight faces was then concatenated with the list of face records produced by the greedy selection.

Thus straight faces never competed against wider faces in the greedy selection, and so the loss of accuracy in the recognition of valid straight faces was avoided. The list of face records was then passed on to the next phase of processing, as in the other version of the greedy selection. Results of the two strategies will be shown below [5.3.4.5, 5.4.2].

5.3.4.3 Enforcing a hierarchy of faces

As observed above, the greedy algorithm removes redundancy of representation and resolves conflicting interpretations of the structure by eliminating overlap between faces. This is achieved by truncation or removal of faces. It may be observed that overlap has been eliminated only amongst faces found at adjacent resolutions; faces further apart in resolution may overlap. This is as it should be, since it reflects the multiple levels of structure in the contour. However, as mentioned above, it is desirable to continue the analysis and build a hierarchical representation of the significant structure: the face tree. This involves the elimination of partial overlap between faces at any resolution. The construction and manipulation of a hierarchical data structure representing the face tree will be considered below [7.1.1]; here the concern is solely the removal of partial overlaps.

A method of quantifying the overlap between faces is a prerequisite of this scheme, and of the greedy algorithm preceding it. Overlap is represented as an ordered pair of real numbers in the range [0,1], say (OA, OB) for a pair of faces A and B. These numbers give the length of the overlap as a fraction of the lengths of faces A and B respectively. If OA has the value 1 it implies that face A is wholly contained within face B; if OB is also 1 then the two faces coincide exactly. If two faces overlap only at their endpoints the overlap values are both taken as zero, since this is not considered a true overlap.

There are many strategies possible for removing the overlap between faces; the decision could be based on face lengths, aspect ratio, strip width, combinations of these, and perhaps other factors. The feature selected in this study was strip width, and in the following discussion the face with the greater covering strip width will be referred to as wider face and the other as the narrower. Four possible ways of resolving the incomplete overlap of two faces have been considered:

(1) delete the overlap from the wider face,

(2) delete the overlap from the narrower face,

(3) break the narrower face into a subface wholly overlapped by the wider and a subface with no overlap, and

(4) extend the wider face to wholly overlap the narrower face.

The possibility of breaking the wider face into two subfaces is not sensible since it would result in a subface wider than the containing face. Of the four possibilities mentioned, the first and the fourth were found to be useful. A consideration of some contour configurations explains this.

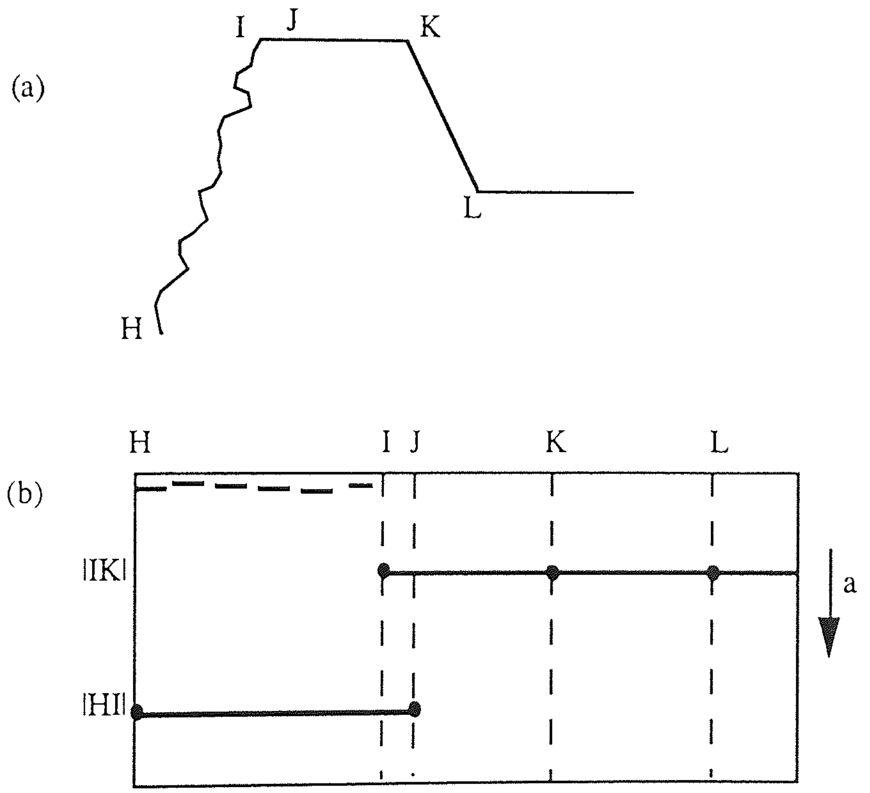


Figure 5.23 Contour segment and sketch of associated face markers in mmspace

Figure 5.23(a) shows a contour segment with a straight face IK abutting a noisier face HI. It should be noted, however, that such a description is making certain assumptions: 'noisiness' cannot be gauged in absolute terms, it must be judged relative to some resolution. In the multiscale analysis used here the noisiness of a contour section is judged by the minimum width for a rectangular strip covering a given arc length. Thus for arc lengths around |IK| in size segments IK and KL are judged smooth, since they produce mmspace values which are small in proportion, and segment HI is judged noisy.

At longer arc lengths, of the order of |HI|, however, section HI yields smaller mmspace values than section IL, because of the relatively short lengths of faces IK and KL and their angular relations. So as the arc length increases beyond |IK|, points on the segment IK near vertex I can have smaller mmspace values than points near K, since the minimal covering strip of the given length will be found to lie along HI and not JK. As a result, the stable zone associated with face HI will extend along IK to some point near J.

Figure 5.23(b) sketches the face markers in mmspace associated with the contour segment of figure 5.23(a). The endpoints of some face markers have been marked by dots. Very short face markers represent the minor straight portions within segment HI found for small arc length values. The face markers for segments IK and KL are found at arc lengths near |IK|; arc lengths around |HI| in value find the face marker from H to J. As is marked in the figure, this results in partially overlapping face markers suggesting faces HJ and IK.

In order to remove this overlap the face marker representing the higher mmspace value is truncated. The result is the pair of faces HI and IK whose overlap is their single common endpoint.

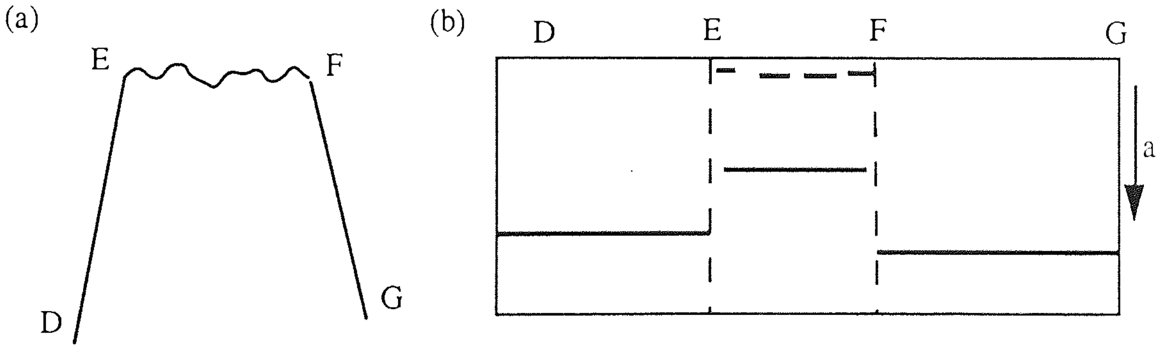


Figure 5.24 Contour section and sketch of associated face markers in mmspace

Figure 5.24(a) shows a slightly different situation where, because of the angular relations of segments DE, EF and FG the segment EF never produces mmspace values smaller than the pair of segments which bracket it. As before, above a certain arc length value, points on EF near E or F may have smaller associated mmspace values than points in the centre of the segment, because their narrowest covering strip extends along DE or FG, and not EF. If sufficiently pronounced, this tapering off of mmspace values may reduce the extent of the associated stable zone and face marker, and produce a consequent failure to find the full extent of the face EF.

Figure 5.24(b) sketches the face markers in mmspace associated with the contour segment of 5.24(a). At short arc lengths the small subfaces of EF may be identified. At longer arc lengths the likely gaps in the representation have been indicated. This shows the likely need for extending the face markers of some faces so as to fully overlap the markers of subfaces

The situations just discussed are to be expected in crystal profile images, which show variable character of faces. This therefore suggests that strategies (1) and (4) above may be useful, and offers no support for strategies (2) and (3). It may be noted that the precise effects seen in mmspace are functions of the relative noisiness of faces, their relative lengths, the angles between faces present, and the arc lengths actually used. It is therefore difficult to analyse these situations, and the choice between outcome (1) and (4) is made on the basis of an empirical rule. The complete procedure for eliminating partial overlap will now be presented; fuller implementation details are given below [5.3.8.4].

Procedure for elimination of partial overlap

The face records are first sorted in order of increasing strip width values. The faces are then considered in turn, starting with the narrowest face, at the top of the list. The face is compared with all faces below it in the list. If no overlap is found then no action is taken. If the narrower face completely or very nearly covers a wider face then the wider face is deleted from the list: an overlap value for the wider face of greater than 0.8 leads to its removal.

If the wider face survives this test then the decision is made whether to truncate it or to extend it to remove the partial overlap. The truncate or extend decision is based on two measures: the difference in direction of the two faces and the fraction of the narrower face overlapped by the wider. The basic idea is that the closer the directions of the two faces, or the greater the degree of overlap, then the likelier it is that the wider face will be extended rather than truncated.

A simple comparison is made between the overlap, expressed as a fraction in the range (0,1], and the angular difference. If the overlap multiplied by 45 exceeds the angular difference expressed in degrees then the wider face is extended; otherwise the wider face is truncated. Following the truncation or extension of the wider face its record is updated accordingly. As before, if the face length becomes too short or the aspect ratio too large then the face will be deleted.

The rule implemented implies that an angular separation greater than 45° will always mean that the wider face is truncated. This may therefore be considered a cautious strategy. Other extend/truncate rules may be applicable, and the current approach is not considered definitive. This simple decision rule appears to work well, however; its performance may be judged from the examples below [5.3.4.5, 5.4.2], where no obvious errors have been noted.

5.3.4.4 Supplementary filter: pruning the face tree

As described above [5.3.3.4], the analysis of some images can produce unwanted subfaces because of the way some oblique lines are realised on a pixel grid as 'staircases' of perfectly straight line segments. An attempt at suppressing the generation of these faces was found to bring unwanted side-effects [5.3.3.4]; the alternative strategy for the elimination of unwanted straight subfaces is to recognise them in the 'final' set of faces and remove them. In contrast to the scheme of modifying the bins of the density slicing, this process would bring no loss of definition in the recognition of valid straight faces. The problem lies in recognising which faces should be removed; the process of removal is straightforward.

Once the need for an additional 'filtering' of the face set to remove unwanted elements had been accepted, it was felt that this opportunity could also be used to remove other faces which seemed likely to be insignificant. It was therefore decided to apply a final 'filter' to try to remove any unnecessary faces, regardless of their origin. It was decided not to use procedures which would only be applicable in the domain of crystal profile analysis. The rules governing the removal of faces were designed to be suitable for any application of the algorithm. Any additional, domain-specific constraints can be applied in a subsequent filtering pass. Possible additional constraints for crystal profile analysis will be considered in the following chapter [7.1.2].

The rules implemented for the removal of unnecessary faces are considered conservative in that a bias has been sought towards keeping an uncertain face rather than removing it. The development of suitable rules is considered to warrant further research. The rules currently implemented are:

1) If a face is very narrow (width < 1.5) or simply very good (aspect ratio < 1/30) then remove all its subfaces.
2) If a subface is perfectly straight or very narrow (width < 0.5), represents less than 15% of the length of a containing face, has a similar direction to the containing face (directions differ by less than 9°) and the containing face is not very wide (width < 4) then remove the subface.
3) If the aspect ratio of a subface is greater than twice that of any containing face then remove the subface.

The first and second rules are designed to recognise the straight subfaces created by digitisation of oblique lines. The second rule will remove subfaces from lines which are not simply digital approximations to straight lines, since the width limit is 4 and the width of a straight line would not be expected to exceed 2, but to limit the power of this rule an additional limit has been set on the fraction of the face which the subface represents. The third rule is similar to the constraint used in Lowe's striptree algorithm for the removal of subtrees, although the use of the factor 2 in the comparison of aspects makes this a much more generous test on the acceptability of faces. Examples of the filtering scheme on real data examples are given below [5.3.4.5; 5.4.2].

5.3.4.5 Illustrative examples: refining the face sets

Synthetic data example

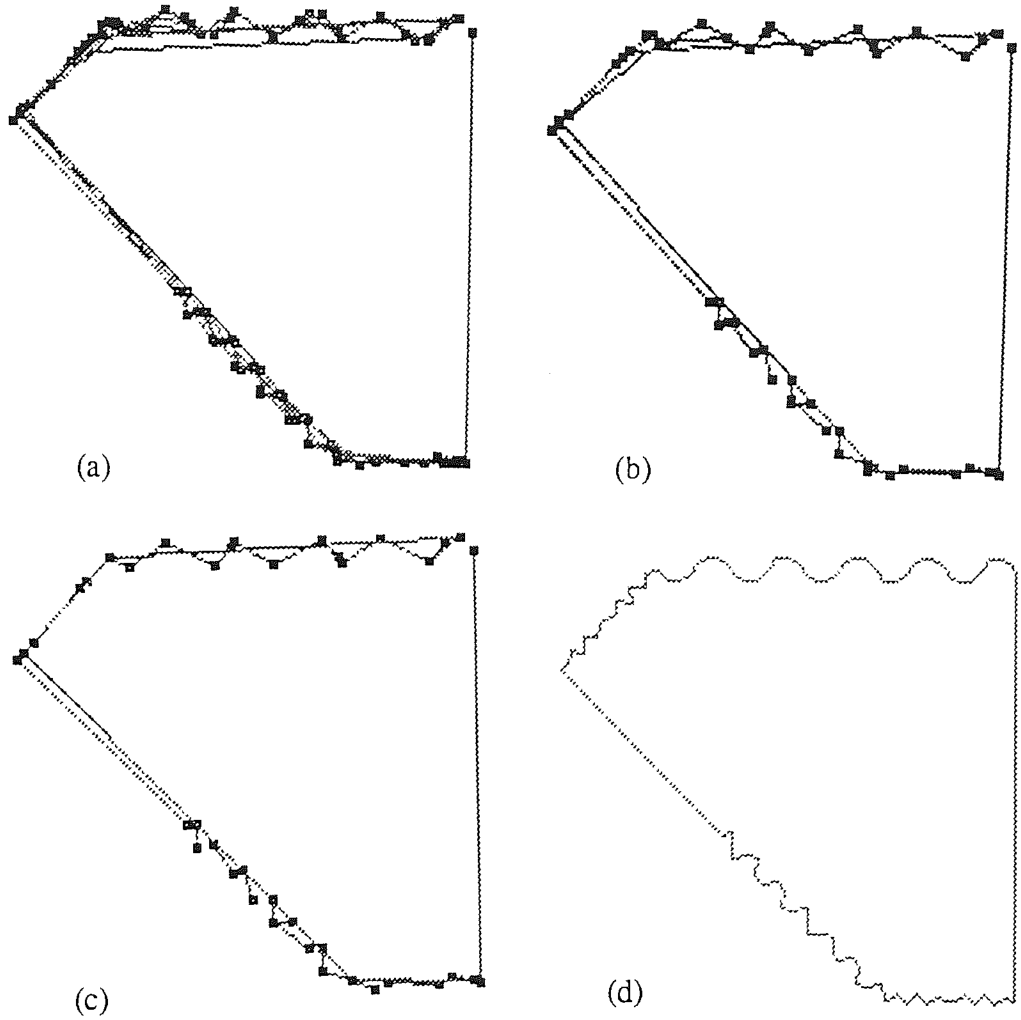


Figure 5.25 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour

Figure 5.25(a) repeats figure 5.15(d) above, and shows all the faces found by the first phase of processing, with the preliminary filter applied to remove faces of aspect ratios greater than 0.33 or lengths less than 6 chainlets. This therefore represents the input to the greedy selection procedure for eliminating redundant faces and choosing a good, representative set. Figure 5.25(b) shows the result of the greedy selection. Figure 5.25(c) shows the data following the endpoint adjustment phase to remove partial overlap among the faces. Figure 5.25(d) reproduces the original contour, for comparison. Again it may be noted that each face is marked by a line drawn between the endpoints found and the resulting line segment need not be the best approximation to the face.

The greedy algorithm can be seen to have been quite effective in selecting a representative subset from the faces presented. The faces found do seem to give useful suggestions of valid structure, and it is only in the irregular portion of the long oblique face that some possible minor subfaces have not been found.

Figure 5.26(a) shows the face markers in mmspace corresponding to the faces of figure 5.25(b); figure 5.26(b) shows the markers corresponding to the endpoint adjusted faces of figure 5.25(c). Figure 5.26(c) shows mmspace, with larger values represented by the darker grey tones. Figures 5.26(a) and (b) show how the faces found are separated in resolution, and clarify the structure where the faces overlap.

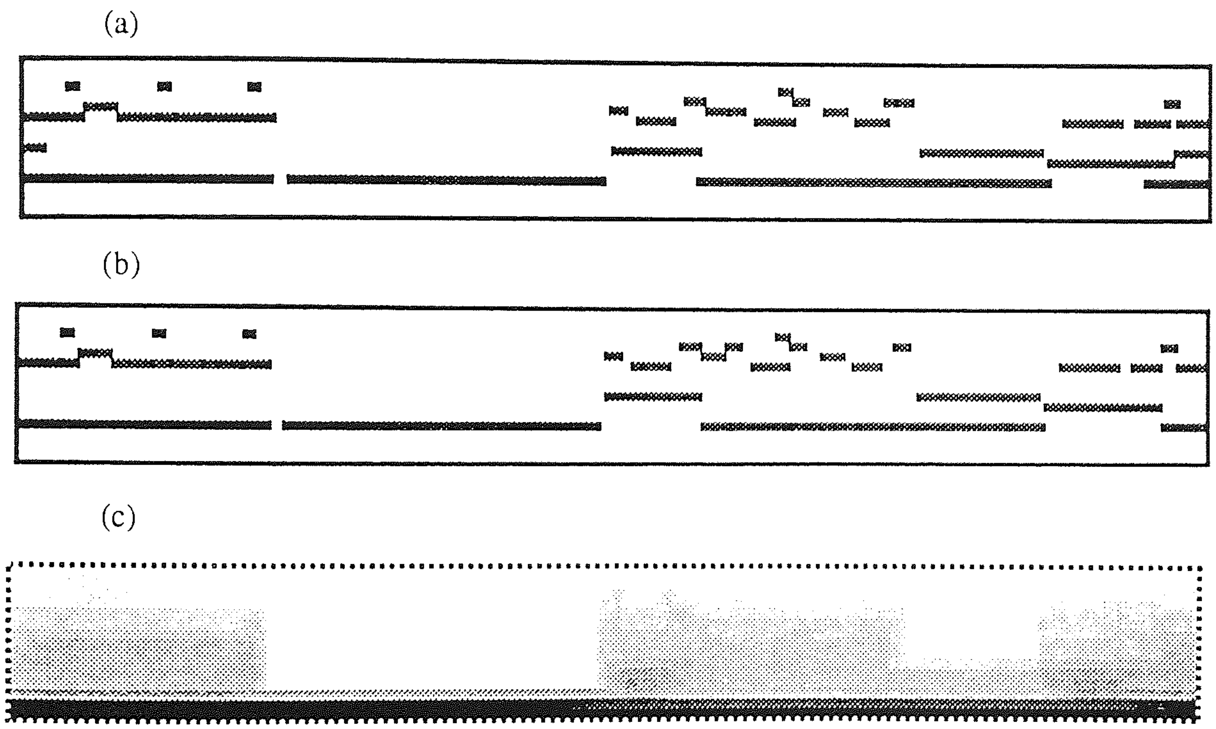


Figure 5.26 Faces markers corresponding to output from greedy selection and output from endpoint adjustment, with mmspace

A comparison of figures 5.26(a) and (b) clearly shows that partial overlap between faces has been eliminated. This has been achieved mainly by shortening faces. One subface at the left hand end of the long irregular upper face has been removed. This came about in two stages. First, the face was truncated since it partly overlapped a narrower face at its right hand end. The shortened face was then found to be overlapped to a large extent by another narrower face. The endpoint adjustment algorithm removes faces which are overlapped by more than 80% of their lengths by narrower faces. The truncated face was therefore deleted.

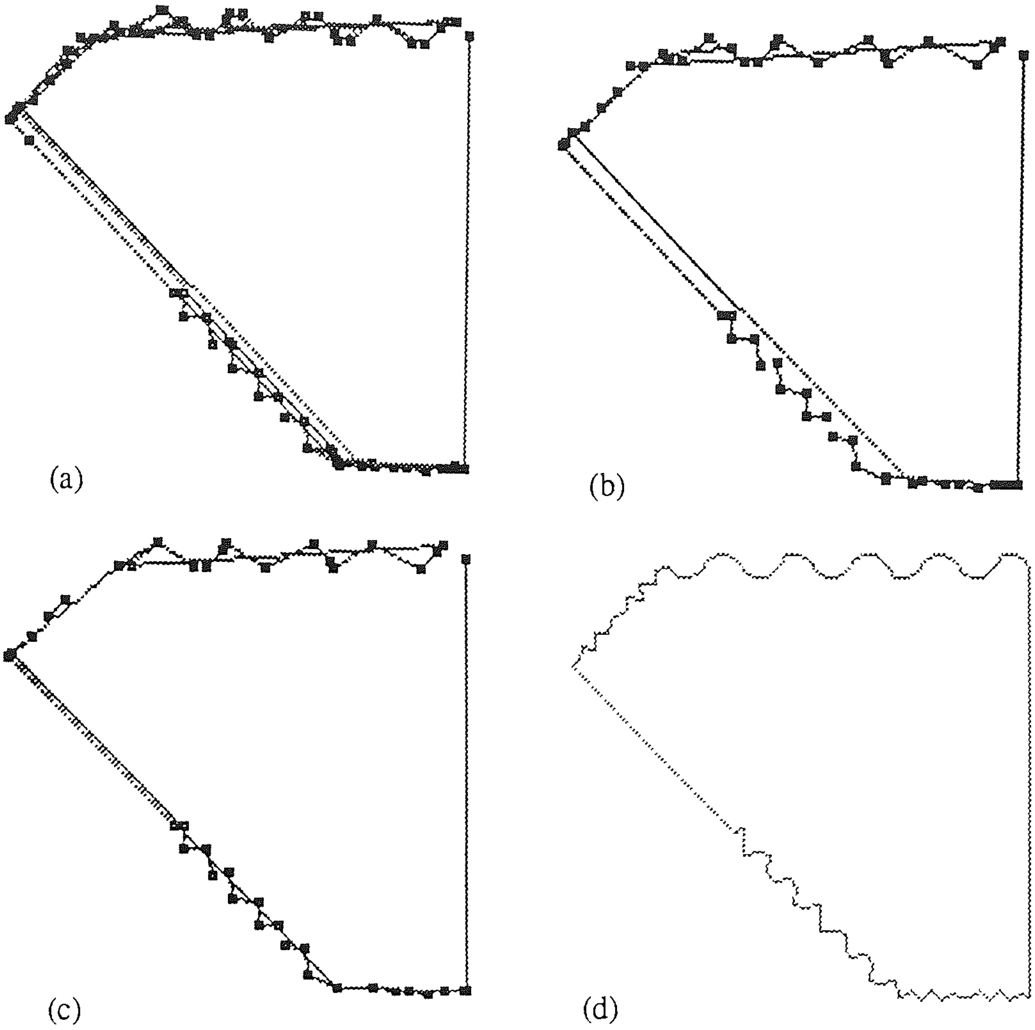


Figure 5.27 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour

Figure 5.27 presents the results of a similar analysis of the faces found for the same profile using the faster arc length progression. Figure 5.27(a) shows the faces found at the end of the first phase of processing, with poor faces removed by the preliminary filter as before. Figure 5.27(b) shows the result of the greedy selection process, and figure 5.27(c) shows the faces after processing to remove partial overlap. Figure 5.27(d) once more shows the original contour, for comparison.

As with the previous data set, the greedy selection and endpoint adjustment procedures are effective in producing a useful description of the possible faces on the contour. An examination of the face markers in mmspace (not shown) was made in order to clarify the structure of the noisy lower face and the oblique upper left face. The lower face was found as one principal face with three small subfaces; the upper left face was found with two minor subfaces.

Topaz crystal profile

Figure 5.28(a) shows all the faces found by the first phase of processing, using the slower arc length progression, i.e. this figure combines the outputs of the three density slicing passes shown in figure 5.19. The preliminary filter has been used to remove faces of aspect ratio greater than 0.25 or lengths less than 6 chainlets. Figure 5.28(b) shows the result of applying the greedy selection algorithm to this data set. It can be seen than more than half of the suggested faces have been removed. The faces which have been retained appear to give a good indication of the likely faces on the contour, and also pick up linear structure at higher and lower resolutions.

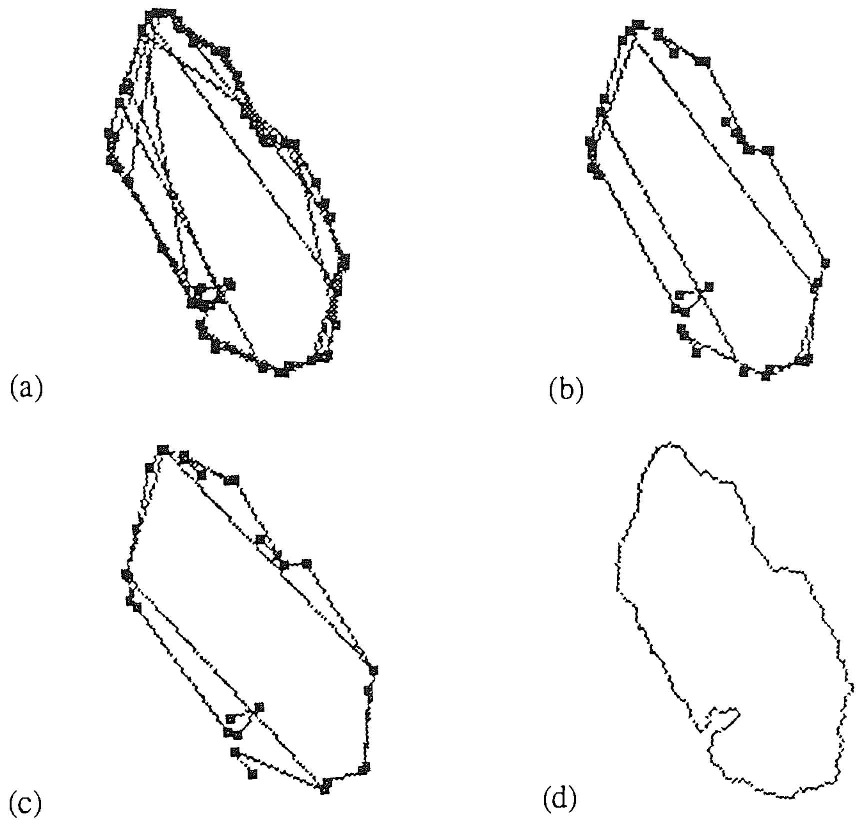


Figure 5.28 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour

Figure 5.28(c) shows the results of processing to eliminate partial overlap between faces: the faces present now represent an implicit face tree structure. Figure 5.28(d) shows the original contour, for comparison. The endpoint adjustment phase has noticibly improved the definition of the two longer faces. As described above, the endpoint adjustment procedure includes a check for pairs of faces representing roughly the same section of contour: if a face found overlaps more than 80% of a wider face then the wider face is eliminated. This rule has led to the removal of the duplication of faces on the bottom left of the figure. Two faces still remain to represent the adjacent side, where the 80% limit has not been reached.

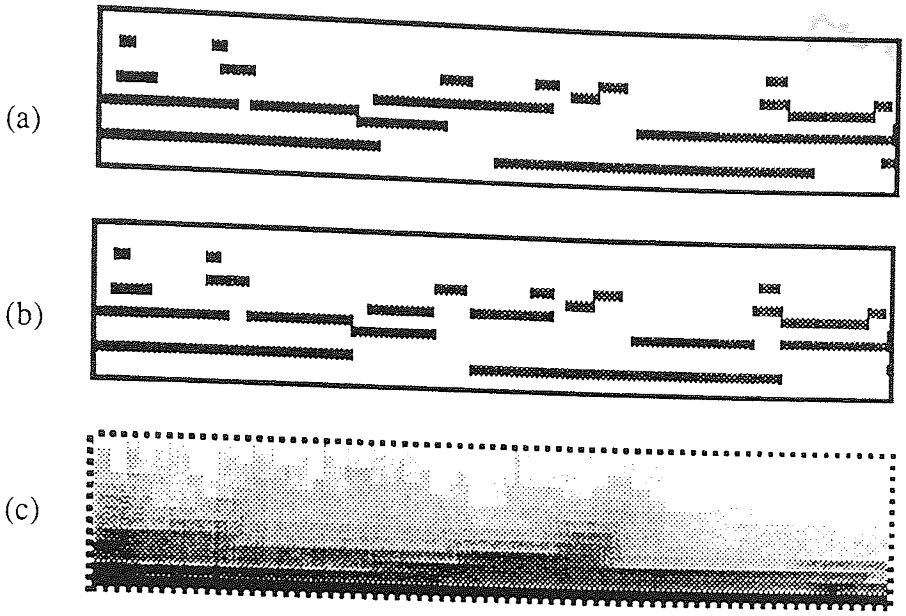


Figure 5.29 Faces markers corresponding to output from greedy selection and output from endpoint adjustment, with mmspace

Figure 5.29(a) shows the face markers in mmspace corresponding to the faces of figure 5.28(b); figure 5.29(b) shows the markers corresponding to the endpoint adjusted faces of figure 5.28(c). Figure 5.29(c) shows mmspace, for reference.

Comparing figures 5.28(c) and (d), the analysis can be seen to have produced a useful description of the linear and near-linear structure on the contour over a range of resolutions. What would probably be considered the likeliest faces have been found, and the suggestion of the two possible longer faces is also reasonable. Their proposal as faces appears more valid when one views the relevant portions of the contour with the rest of the contour hidden. The symmetry in this figure acts strongly to support the selection of the sides of smaller aspect ratio. Since it is perfectly possible that another arrangement of faces could be present where the coarser faces form the more symmetric figure and thus represent the better final choice, the faces should be retained. The final selection of faces may then be made on the basis of the symmetry properties of the overall figure.

Figure 5.30 presents the results of continuing the analysis of the faces found using the faster arc length progression. The processing is otherwise identical to the previous example. Figure 5.30(a) shows the faces found at the end of the first phase of processing, with poor faces removed by the preliminary filter, as before. Figure 5.30(b) shows the result of the greedy selection process, and figure 5.30(c) shows the faces after processing to remove partial overlap.

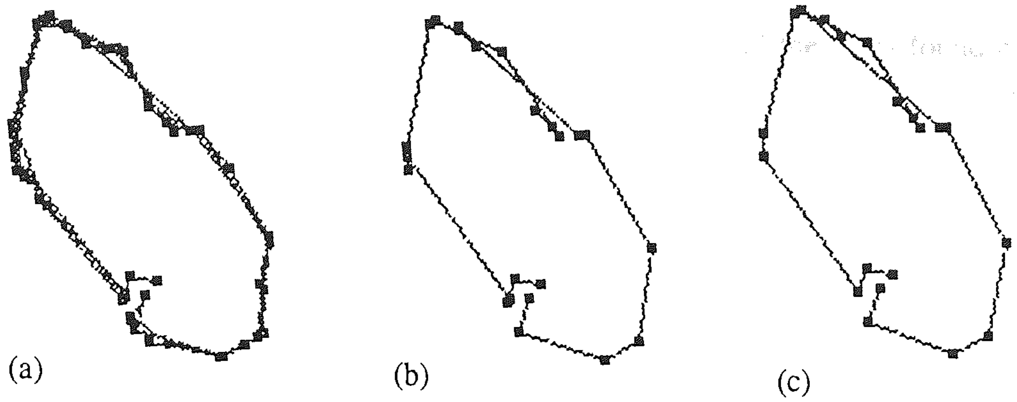


Figure 5.30 Faces input to greedy selection, output from greedy selection, and output from endpoint adjustment

Because of the change in the initial processing phase, compared with the previous analysis, less suggestions for possible faces have been made. However, the analysis appears to have again found all the more important faces, but with fewer suggestions for lower resolution structure. Only one relatively coarse face has been found, on the figure top right - a possible face not found in the previous analysis.

Lamprophyllite crystal profile

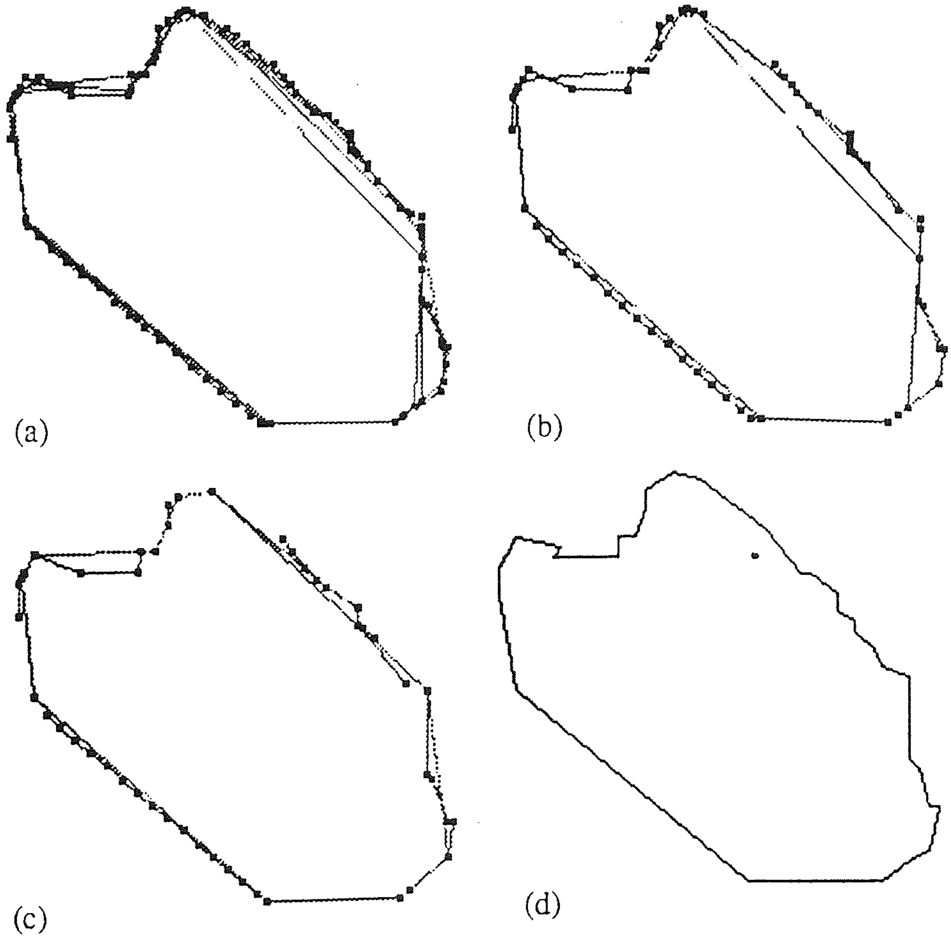


Figure 5.31 Faces input to greedy selection, output from greedy selection, output from endpoint adjustment, and the original contour

Figure 5.31(a) repeats figure 5.22(a), and shows all the faces found by the first phase of processing, using the slower arc length progression, and with the preliminary filter applied to remove faces of aspect ratio greater than 0.25 or of length greater than 6 chainlets. The density slicing retained the special category for faces of zero width, and this has led to more than a dozen small step faces being picked out on the long straight oblique face.

Figure 5.31(b) shows the result of the greedy selection of faces applied to this data set. This has thinned out the set of possibilities while retaining a good representative subset. Figure 5.31(c) shows the result of the endpoint adjustment procedure which removes partial overlap between faces. Figure 5.31(d) shows the original contour, for comparison.

The endpoint adjustment procedure appears to have been very effective in constructing a hierarchy of faces. The estimate for the long, slightly irregular face on the top right of the figure has been trimmed and now delimits the face very well. Similarly, the estimate of the irregular face on the right of the figure, with three relatively straight subfaces, has been much improved. A small but relatively straight face which can be seen on the top left of the original contour has not been found, and the face suggested for this segment is too long. One of the density slicing passes did pick up at least part of the face, as can be seen in figure 5.31(a), but the greedy selection procedure rejected it in favour of the longer, less accurate face estimate which was found using the next longest arc length.

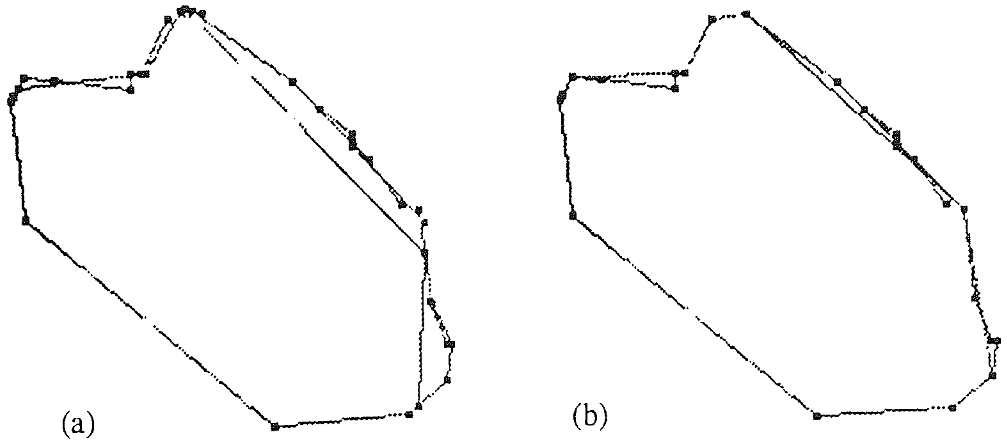


Figure 5.32 Faces output from greedy selection, and faces output from endpoint adjustment

All the subfaces of the long oblique lower left face still survive the processing. Figure 5.32 shows the processing of the image when the generation of such faces was suppressed by excluding the zero-width category from the density slicing. Figure 5.22(b) above showed the faces produced by the first phase of processing. Figure 5.32(a) shows the result of the greedy selection and figure 5.32(b) shows the faces following endpoint adjustment to remove partial overlap. As noted previously [5.3.3.5], this method has eliminated the subfaces of the long oblique face, but has produced poorer estimates of the longer straight faces on the contour.

Figure 5.33 illustrates the alternative and preferred strategy for improving the final set of faces, namely recognising and deleting unnecessary subfaces. For this example the slightly modified processing route described above [5.3.4.2] was used: the perfectly straight faces were not submitted to the greedy selection, and the density slicing passes to find the faces of non-zero width did not use the zero-width category. Figure 5.33(a) shows the faces produced by applying the greedy selection to the faces of non-zero width found in the first phase of processing, with the perfectly straight faces added afterwards. This figure should therefore combine the best features of figures 5.31(b) and 5.32(a).

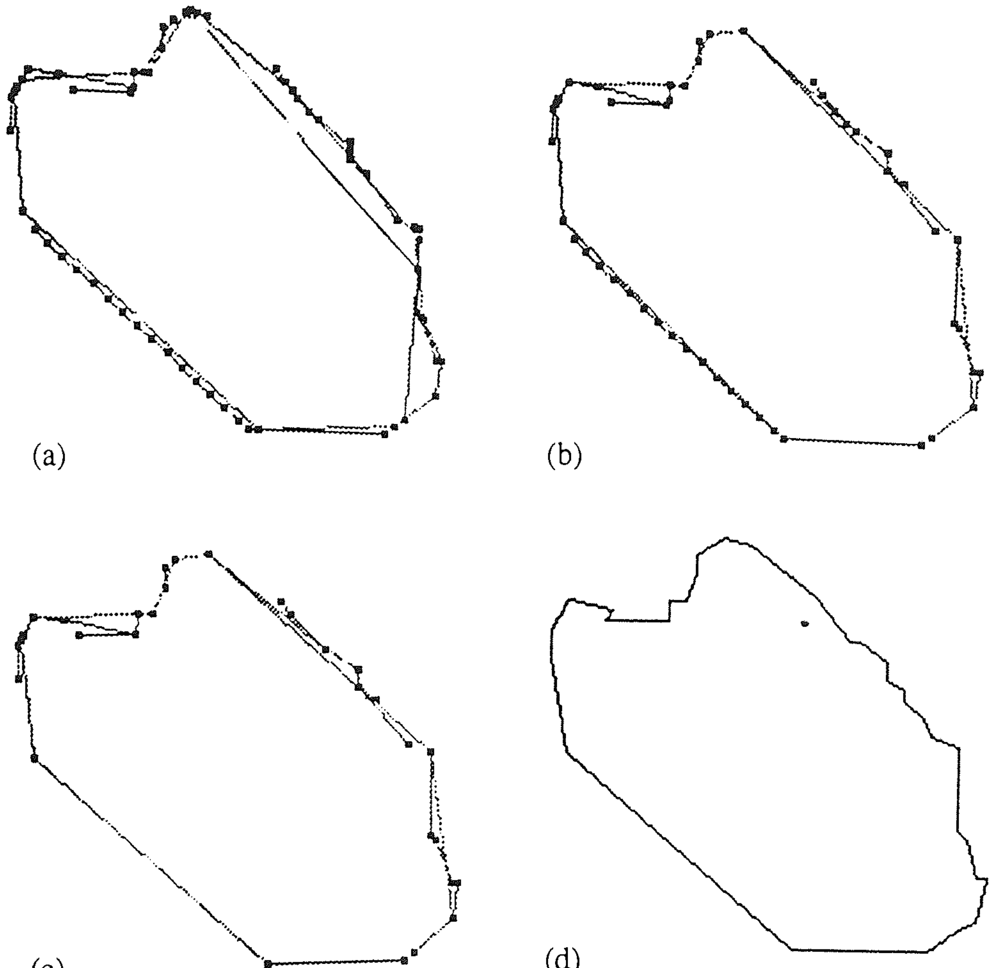


Figure 5.33 Faces output from greedy selection, faces output from endpoint adjustment, faces following the supplementary filter, and the original contour

Figure 5.33(b) shows the result of processing to remove partial overlap between the faces. Figure 5.33(c) shows the result of the supplementary filter to remove unwanted subfaces. This can be seen to have been completely effective in removing the 'step' subfaces of the long oblique face. Some unnecessary subfaces of the long upper face have also been removed, although a few remain, but in comparison with figure 5.32(b) it can be seen that the straight faces are better defined. The final result is considered to be a good, useful description of the important linear structure on the contour.

More examples of the analysis of these contours will be given below [5.4.2].

5.3.5 Generating limited error polygonal approximations

The complete algorithm for multiscale analysis described above produces as its output a list of face records describing what has been found to be the significant linear structure of the contour for a range of resolutions. The list of face records can be used to construct a face tree for the contour which reflects the hierarchy of structure in a natural manner. The primary aim of the analysis was to find and describe all significant structure on the contour. Accordingly, little attention has been paid to amorphous contours or portions of contour. However, there is also a need for a representation for the entire contour of a crystal profile, to provide data compression or simplification for subsequent processing steps such as convex hull computation and area calculation. What is required, therefore, is an additional polygonal approximation of the more 'traditional' type, i.e. one where the same error criteria are used at all contour points.

The techniques described above [5.2.1] could be used to provide a conventional, limited error polygonal approximation. However, it is also possible to generate such an approximation from the intermediate representation, as will now be described. The method is not proposed as a rival to the algorithms discussed above, since the intermediate representation is relatively computationally expensive to generate. However, given that this representation is already available, as would be the case in crystal profile analysis, the required polygonal approximation can be produced with little additional processing.

The process of finding a limited error polygonal approximation effectively requires the face tree to be developed downwards (i.e. away from the root) until each face satisfies the width limitation, and every point on the contour is part of some face. Thus the face tree would be extended to provide a complete tessellation of mmspace analogous to figure 5.3. In practice no explicit tree structure need be produced, however.

In accordance with the discussion on arc lengths [5.3.3.1] the assumption is made that the minimum arc length used should produce mmspace values of zero at all contour points. An arc length of 1 need not be considered since it could provide no useful information; the minimum possible arc length used is 2, which may produce non-zero mmspace values.

The algorithm proceeds in a manner similar to that discussed above [5.3.3.3]. Only one threshold is used: the error of fit specified by the user. This produces a binary array such as that illustrated in figure 5.9(b), for example. Zone markers delineating the tops of the 'steps' in the thresholded array are then extracted. If the zone marker indicates that the shortest arc length has produced non-zero mmspace values then the required polyline for this part of the approximation can be achieved by placing breakpoints only one chainlet apart within the limits of the zone marker.

For zones found at other arc lengths the approach described above for zone analysis is followed. In order to correctly handle gently curving portions of the contour, however, the permitted separation of breakpoints is reduced, compared with the face finding analysis. Also the practice of eliminating the search window in cpspace around breakpoints is adapted to allow adjacent breakpoints. Figure 5.34 illustrates the need for this step when a zero error threshold is specified; analogous situations may arise for non-zero thresholds.

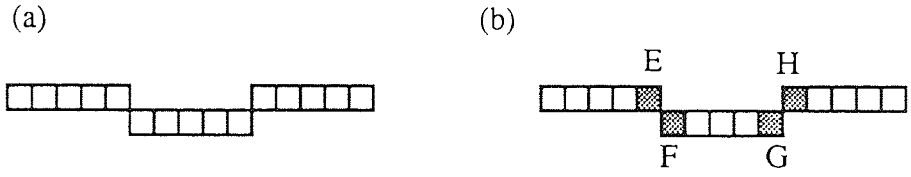


Figure 5.34 Contour segment requiring double breakpoints

Figure 5.34(a) shows a contour segment which could be covered by strips of zero width for arc lengths up to 4, but would give non-zero mmspace values for longer arc lengths. If a zero error fit were specified, a perfect fit would require the use of double breakpoints, at positions E and F, and at G and H, as shown in figure 5.34(b). The previous approach of eliminating the search window around each corner found is modified to cater for this situation by not eliminating the search window at the two immediate neighbours of a corner. Thus on subsequent passes of the search the other corners in the pair will be found. The need for double breakpoints was ignored in the earlier analysis since the small loss of accuracy was considered insignificant.

Figure 5.35 shows the approximations generated by the resulting algorithm. Figure 5.35(a) shows the original test contour; figures 5.35(b) and (c) show superimposed the approximations for errors of 2 and 4. For comparison, figures 5.35(d) and (e) show the results of a limited error polygonal approximation algorithm due to Williams [1978], with error tolerances of 2 and 4 respectively.

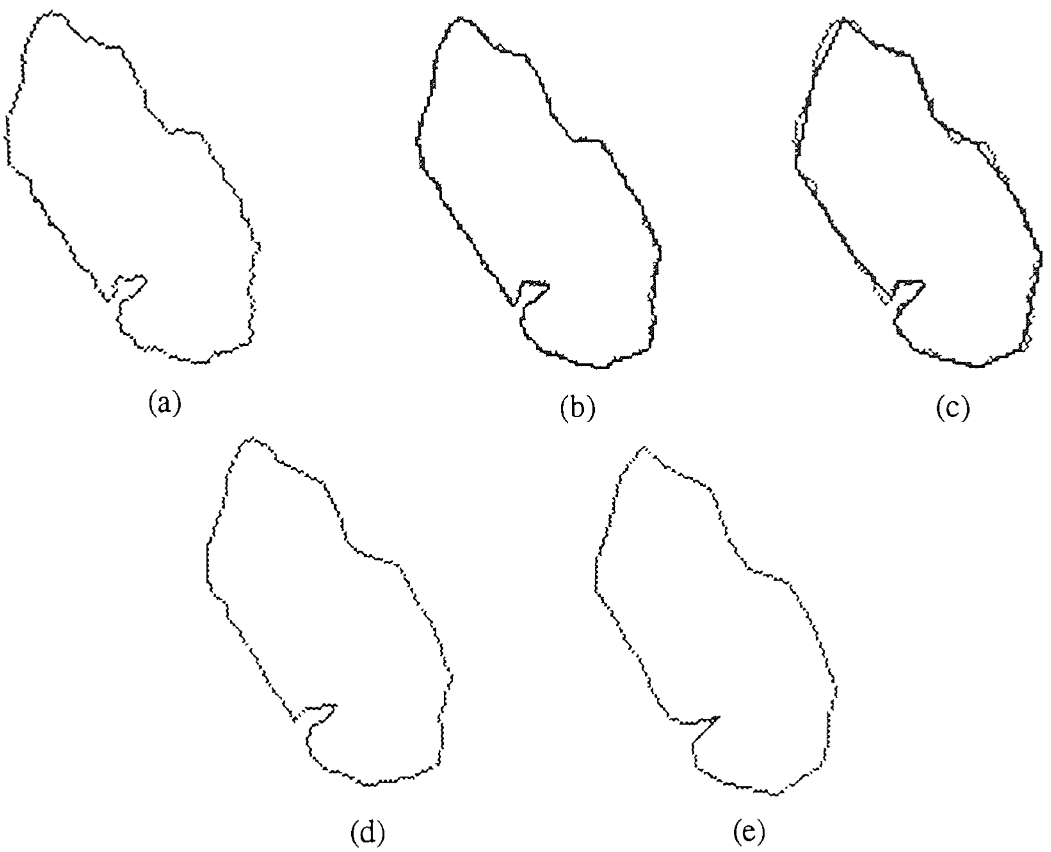


Figure 5.35 Fixed error approximations to a contour

It is a requirement that the polygonal approximation should complement any possible set of faces produced by the multiresolution analysis, by supplying a description of the remainer of the contour. This may be achieved by traversing the list of records for the faces found and inserting their end points in the list of vertices which defines the fixed error approximation.

5.3.6 Development of the algorithm

Sections 5.3.1 to 5.3.4 above, which discuss the creation of an intermediate representation for contours and how this can be processed to produce a useful description of the linear structure on the contour, broadly reflect the course of development of the analysis algorithm. The development history will now be reviewed in greater detail. This will be followed by a description of the current form of the algorithm. This description will concentrate on how the algorithm is used; implementation details of the algorithm will be given in section 5.3.7. The question of evaluating the algorithm's performance will be considered in section 5.4. Possible future development of the algorithm will be considered in section 5.5.

5.3.6.1 History of development

The need for a new algorithm was recognised during the investigation of existing shape analysis methods. Davis' ADS algorithm and the Phillips and Rosenfeld corner detection scheme were implemented. Following the strategy used in ADS, the Phillips algorithm was applied for a series of arc lengths, creating the two dimensional array which was later termed 'cpspace'. With Davis' SDS algorithm in mind a way was then sought to derive complementary information on the contour 'sides'. The idea of applying a 'minimax' principle to the arc-chord values was then devised, and 'mmspace' was created. This particular representation of a contour is novel.

After experimenting with the properties of 'cpspace' and 'mmspace', by generating images of these arrays for various test contours, it was felt that these arrays together carried enough information from which to derive the required structural description. 'Stable zones' were visible in mmspace and clearly represented portions of the contour corresponding to possible faces; the major corners and minor corrugations on the contour could be related to ridges seen in cpspace. The nesting of the stable rectangles was also clear, and seemed to provide a natural description of the hierarchical arrangement of the faces. The face tree which could be derived was seen to have analogies with Witkin's interval tree description of one-dimensional curves, and seemed to be a more direct representation of contour structure than Lowe's binary strip tree. The principal problems to be addressed were then: (1) how to recognise and extract the relevant information from the stable zones in mmspace, and (2) how precisely to relate the corner and side information.

The strategies tried for extracting stable zones have already been described [5.3.3.2] The idea of density slicing was eventually settled upon. A description is given below [5.3.8.1] of the technique used for extracting the stable zones. Zone markers were at first represented as line segments drawn in an array of dimensions exactly corresponding to those of mmspace, i.e. zone data was not explicitly represented in records. Initially only one set of thresholds was used in the density slicing; considerations in selecting threshold sets have been detailed above [5.3.3.3].

The problem of breaking the zones up into faces was then considered. The cpspace values clearly provided the sort of information required to perform this. After consideration of the nature of mmspace and cpspace the ideas arose of using mmspace values to define an acceptance threshold on the corner values, and using the adjacent arc length to supply the corner information [5.3.3.4]. The need to find local maxima in sectors of cpspace led to the practice of finding all local maxima at the start and storing their positions in a separate array. Restricting a search window could then also be implemented by deleting the relevant local maxima positions.

At this stage in the development experimentation was used to refine the choice of parameters in the algorithm. However, for the reasons given above [5.3.3.3], it became clear that a single set of thresholds could not extract the required structural information in some situations: this is the problem of variations in optimal resolution around a contour. A solution to this problem proved difficult to find but was nevertheless straightforward: use several threshold sets and then combine the various interpretations which result.

Once the need for several thresholds had been accepted the idea of designing sets specifically to capture likely zones was considered, as was reported above [5.3.3.3]. However, the simpler solution was adopted of using three sets which gave rise to overlapping intervals or 'bins'. (Later, five threshold sets were also tried, but did not significantly improve face identification.)

Problems in the use of line segments marked in arrays to represent face data had been noted earlier, but it also became clear at this stage that this representation could not readily support the sort of further processing which was envisaged. The program was therefore revised to use record structures for the stable zones and possible faces.

A method had then to be devised for combining the structural interpretations produced by the three threshold sets: taking the best faces, removing redundancies and resolving ambiguities. A 'greedy' strategy of some sort was considered to be both simple and appropriate to the problem.

The first greedy scheme implemented operated on the set of all face records and used both length and width information. Faces were removed which were overlapped to at least 75% of their extent by a longer face of similar width. This process aimed to retain faces at different resolutions, since overlapping faces were only allowed if their widths were significantly different. The procedure appeared to be quite effective on the examples tested, but involved large numbers of comparisons and tests for overlap. Additional weaknesses were its reliance on width values which were only approximations to the true values, and the problem of defining what a significant width difference actually was. (A method for improving the face width estimate was developed much later.)

The greedy selection procedure outlined above [5.3.4.2] was then devised. The algorithm is described in detail below [5.3.8.3]. At first the selection process was applied only for sets of face records generated using the same arc length value, i.e. the candidate faces for each row of mmspace were treated in turn. It was then realised that the faces could be grouped in sets representing two adjacent rows of mmspace, i.e. two successive arc lengths. This produced a final selection of faces where any overlap should reflect structure found at different resolutions.

An end point adjustment phase was still required in order to remove partial overlap between faces found at different arclengths. An examination of contour configurations showed a need for face extension in some situations and a need for truncation in others. The first extend / truncate decision making algorithm which was implemented used a simple empirical rule based only on the relation between the overlap expressed as a fraction of the first face involved and the overlap expressed as a fraction of the second. This rule proved inadequate, occasionally causing the loss of good faces through inappropriate truncation. A second algorithm was then devised which used both overlap and relative face direction information. The algorithm devised to perfom this has been outlined above [5.3.4.4] and is described in detail below [5.3.8.4].

Experimentation with the algorithm at this stage, using hand digitised real data examples, disclosed how the algorithm found the small perfectly straight segments by which oblique lines are realised on a discrete pixel grid. One strategy tried for eliminating these faces was to modify the density slicing by removing the category for faces of zero width. As shown above [5.3.3.5], this change brought unwelcome side effects. However, it did suggest one modification which was made an option in the analysis: using the adjusted density slicing scheme plus one processing pass to extract the straight faces, and combining the results after greedy selection had been used on the other faces [5.3.4.2].

Since suppressing the generation of the unnecessary faces had its drawbacks the direct approach of identifying and deleting them was used. The procedure for recognising unwanted faces has been described above [5.3.4.4].

5.3.6.2 Current form of the algorithm

The multiscale analysis program produces descriptions of the linear or near-linear structures on a two-dimensional profile. The descriptions are presented in two ways. Final and intermediate results are presented in the form of image files which may be displayed for visual inspection. The program also produces a data file in which are tabulated the field values of the face records of the required set of faces. The program has not yet been incorporated into the integrated analysis scheme discussed in chapter 4: the program stands alone, accepting input in the form of a binary image and producing its results as images and data files. Strictly speaking, there are two versions of the program since the processing modification whereby straight faces are excluded from the greedy selection [5.3.4.2] was implemented as a separate program and not as an option made available during processing.

The first step in the program is to scan the input image to locate the first non-zero object and generate the chain code of the contour of this object. It would therefore be very simple to alter the program to accept directly a chain coded contour description. The chain code may describe a contour which is 4-connected or 8-

connected [4.1.1]. One restriction on the form of the contour is that it should have no 'necks' (isthmuses) or projections of only one pixel thickness, i.e. points which appear more than once on the contour, since this may lead to a chord length of zero being found and produce divide-by-zero errors in the calculation of the intermediate representation.

The program is used interactively, and is run initially with the input file name as the sole argument. The user is then prompted to give the x and y dimensions of the image array. The image file is read in and the contour chain code calculated. The user is then asked to specify the arc length progression to use in the analysis (i.e. faster or slower [5.3.3.1]).

The program can be set to offer many intermediate displays which have been found useful in examining the state of the processing. It is at this point that the user is asked if he would like to be offered the full range of possible displays, or only a small subset. The user then enters the limits on acceptable faces [5.3.4.1]: the maximum acceptable aspect ratio and the minimum acceptable face length. No more parameters are set by the user.

If the full gamut of intermediate displays has been requested the user will be offered, for example, display files of mmspace and the fixed error polygonal approximation. Further, on each of the three density slicing operations, the following will be offered: display files of the stable zones on the contour, of zone markers in mmspace, listings on screen of the zone data, display files of the possible faces on the contour, of face markers in mmspace, and listings on screen of the face data.

Regardless of the options previously selected, the user is then offered an image file of the combined sets of possible faces (e.g. figure 5.25(a)), an image file of the faces found by the greedy selection (e.g. figure 5.25(b)) and an image file of the corresponding face markers in mmspace (e.g. figure 5.26(a)). Similarly, the user is offered an image file of the faces following endpoint adjustment (e.g. figure 5.25(c)) and an image file of the corresponding face markers in mmspace (e.g. figure 5.26(b)). In addition the user is given the option of storing a tabulation of all the face record data to a named filed. The same options are offered after the final processing phase which aims to remove poor or unnecessary faces from the face tree (giving, for example, figure 5.33).

5.3.7 Detail of the algorithm

The algorithm for multiscale contour analysis outlined above [5.3.3 to 5.3.6] has been implemented in C on an HLH Orion 1/05 mini-computer. The executable code for the program occupies 76K. This figure should be taken as a very generous indication of the space required, however, since the current implementation contains many extra facilities for the production of intermediate displays and the testing of different options.

Two principal types of compound data structures are used: arrays, both one and two-dimensional, and a record structure which can be used to form linked lists. The two two-dimensional arrays of the intermediate representation, cpspace and mmspace, are of size lcode X narcs, where lcode is the chain code length and narcs is the number of arc lengths used in the multiscale analysis. In addition, similar arrays are used to store the locations of the local maxima in acspace (array 'locmax') and the total of the maximum left and right deflections of the arc from the chord for every arc length used, at every position on the contour (array 'arcmax').

In the descriptions of the algorithms given below the arrays will be shown as conventional two-dimensional arrays. However, they are actually implemented as one-dimensional arrays which can be dynamically allocated using the C 'calloc()' function. The program is therefore flexible in that arrays of variable size can be created when required.

The chain code of the contour is stored in a simple array. Two similar arrays, xco and yco, record the x and y coordinates of the corresponding contour points. The direction of the chain encoding gives an ordering to the contour points and this allows reference to be made to the startpoint and endpoint of faces and zones on the contour.

The zone records and face records are implemented as C 'struct' data types. The same record structure is used for both. These records take the form:

```
struct face_rec {
     int    startch, endch, pseudolen, xstart, ystart, xend,
            yend, arcused;
     float  truelen, maxwid, aspectrat, fdir;
     struct face_rec    *link;
     }
```

The named fields have already been described [5.3.4]. Briefly, startch and endch are the chainlet positions at the start and end of the face; xstart, ystart, xend and yend are the coordinates of the endpoints; pseudolen is the number of chainlets on the contour between the face endpoints; truelen is the Euclidean distance between the face endpoints; arcused indicates the arclength used to find this face, specified as the index in an array of arc lengths; maxwid is a measurement or estimate of the 'width' of the face, i.e. the width of a rectangle needed to cover it; aspectrat is the 'aspect ratio' of the face, being the face width divided by the truelen value; fdir is the direction of the face normal, given in degrees; link is a pointer used to create linked lists of records.

The high level structure of the program is straightforward, and uses a simple processing sequence, with one processing step iterated three times. First the intermediate representation of the contour is created. Next possible faces are found using three density slicing passes. The combined list of records for the possible faces is then reduced to select the best subset of faces without redundancies or contradictions. Partial overlap between faces is eliminated to produce a strict hierarchy of faces. Poor or unnecessary faces can then be sought and deleted.

The generation of the cpspace and mmspace arrays has already been discussed in detail [5.3.1]. The other principle stages of the program which have been identified are considered here in three sections: creating the zone and face records, the greedy selection of one set of faces from a list of candidate faces, and face endpoint adjustment. In addition, the implementations of subroutines for ordering and overlap are considered in a separate section.

Familiarity is assumed with some basic elements of C notation such as the increment (++) and decrement (--) operators and the `while` and `for` loop structures which will be used in the algorithm descriptions below. It may also be useful to note that, as in C, arrays of dimension N are indexed with the values 0 to N-1.

5.3.7.1 Producing the lists of zone and face records

Three passes are made over the intermediate representation, one per threshold set. On each pass the stable zones are identified, a list of zone records created, possible faces identified, and a list of face records created. The three lists of face records are concatenated to produce the final linked list of face records which will be pruned and refined in subsequent stages of processing.

Finding zones

The first step on each processing iteration is to set up the array of thresholds for use in the density slicing. The thresholds are specified by their first non-zero step and the recurrence rule for setting the successive threshold step given the previous step [5.3.3.3]. The threshold levels are stored in the array 'thsteps'. thsteps[0] is set to zero and thsteps[nosteps-1] is greater than any data value found in the mmspace array.

The density slicing of mmspace can then be performed. This produces a new array 'dsmm' which is the density sliced mmspace. The core of the density slicing loop is:

```
level = 0;
while (mmspace[x][y] > thsteps[level]) level++;
dsmm[x][y] = level;
```

The dsmm array can thus be seen to hold integer values in the range [0, nosteps-1] , since the index of the threshold value in the array thsteps is recorded, not the threshold value itself. This simplifies the search for stable zones in dsmm.

A new array 'tempspace' in registration with dsmm is used to store the zone markers as line segments prior to the creation of the record structures. The stable zones are identified for each threshold interval in turn:

```
for (index_value = 0; index_value < nsteps; index_value++)
          {
          Find the stable zones.
          Create zone records and add them to the linked list of records
          }
```

A preliminary step in the process of finding the stable zones is to clear the tempspace array in which the zone markers will be represented. It may be recalled that zone markers are the lower edges of the stable rectangles in mmspace [5.3.??] (using the convention that the lower edge of a rectangle in mmspace is the edge with the *greater* y-coordinate). The rows of dsmm are then analysed in turn. If the array element dsmm[x][y] has the index value currently being sought, and dsmm[x][y-1] also has this value, and dsmm[x][y+1] has a greater value, then the point (x,y) may be on the lower edge of a stable zone, and tempspace[x][y] is set to 1.

For the top (y=0) and bottom (y=narcs-1) rows of dsmm the tests must be modified. The array element tempspace[x][0] is set to 1 if the corresponding element of dsmm holds the index value currently sought and dsmm[x][1] has a greater value. The array element tempspace[x][narcs-1] is set to 1 if the corresponding element of dsmm holds the index value currently sought and dsmm[x][narcs-2] also has this value.

When every row of dsmm has been considered in this way using the current index value tempspace will be complete. tempspace may then be processed to convert the zone markers into zone records.

tempspace is analysed row by row. The elements of the row are first summed. If the sum is zero then there are no zone markers on the row. If the sum equals lcode, i.e. the length of the row, then the whole row is a stable zone. It is then necessary to find a suitable beginning and end point for the zone to store in the zone record. The corresponding row of acspace is therefore examined and the position of the largest corner value is found. This position is taken as a suitable startpoint for the zone and the adjacent preceding point is taken as the zone endpoint. A zone record is then created and added to the linked list of records for this threshold set.

If the row sum is neither zero nor lcode then there may be one or more distinct zone markers on the row. The markers are found by tracking along the row. For each zone marker encountered a zone record is created, the relevant data are stored, and the record is added to the linked list.

When this process has been repeated for each row of tempspace, for each threshold array index value, the linked list of zone records will be complete for the current threshold set under consideration. The next stage of the analysis may then be started: finding the faces represented by these zones.

Identifying possible faces

The linked list of zone records is traversed and each record is analysed. During this process a new linked list of the corresponding face records is constructed. For each zone the zone width and endpoints are recovered from the zone record, as well as the number of the row in mmspace on which the zone was found. This information will be used to set up a search window on cpspace where the corner values will be examined in order to estimate how many contiguous faces the zone represents, and their limits.

The zone width value stored is the thsteps[index] value for the zone, i.e. the upper limit of the density slicing 'bin' used to find this zone. The zone width will be used to judge the significance of the size of the corners found within the zone. A fairer estimate of the true zone width is to take the mean of the upper and lower limits of the density slicing bin. Since the upper limit is always twice the lower limit, except when the lower limit is zero, the mean zone width is estimated by taking three quarters of the width value stored in the zone record.

The corner values in cpspace are examined on the row adjacent to the row on which the zone was found, i.e. at the next greater arc length. The appropriate row of cpspace can therefore be selected by referring to the zone record data, and the zone endpoints used to define a search window on this row. This window is further reduced by bringing in the endpoints, to reduce the influence of possible corners outside the zone. In practice, the reduction used with the faster arc length progression is the arc length corresponding to the row on which the zone was found, less 2; for the slower arc length progression 7/10 of the arc length was used.

The positions of the local maxima in cpspace have all been found previously, following the generation of the intermediate representation, and the maximal positions marked by 1's in the array 'locmax'. A copy of this array, 'lmax', is used to assist the search procedure. The search window is set up on lmax and at the non-zero positions within this window the corresponding acspace values are examined. The size of the maximum corner found in the window is compared with the average zone width value. If the corner size is less than this threshold then the search for corners within this zone terminates. If the corner size is greater than this threshold then the corner position is marked in another array, 'bkpt', by setting the appropriate location to 1. bkpt has length lcode and is initially set to zero. The search window must then be reduced so that no more corners are sought in the immediate neighbourhood of a previously found corner. Using the lmax array to indicate the points of cpspace which should be examined allows the restrictions on the search to be implemented easily: the lmax array is set to zero in the neighbourhood of the corner found, and at the corner point itself. In practice, when the slower arc length progression was used a value of 3/4 of the arc length was used as the radius of the neighbourhood to be removed from further examination around each corner; for the faster arc length progression the fraction used was 7/8.

Eventually no corners above the threshold size will be found in the increasingly blanked off window, and the search terminates. The corner positions recorded in the bkpt array must now be used to generate the face records. The bkpt array records the ends of the faces found strictly within the stable zone. Thus the presence of n nonzero points in bkpt indicates that the zone contains n+1 faces.

The first step in analysing bkpt is to count its zonzero elements. If this value is zero then a single face record is created, the data fields from the zone record are copied across without alteration, and the record is added to a linked list of face records. If the number of nonzero elements in bkpt is one or greater then the new face endpoints are identified, face records are generated in turn and added to the linked list. The zone start point is used as the start point of the first face in the zone and the zone end point is the end point of the final face in the zone.

This process is carried out for each zone marker on each row of tempspace. This results in a linked list of records for all faces found using this particular threshold set. This list is than concatenated with a 'master' list which, at the end of the three processing passes, represents all the possible faces found.

The face records carry enough information to allow this list to be reduced by the removal of 'poor' faces, i.e. the records of those faces which are shorter than a pre-set threshold or whose aspect ratios are greater than a pre-set threshold. The reduced linked list may then be passed on to the next stage of processing which is the 'greedy' selection algorithm.

5.3.7.2 Finding face overlap and sorting face records

The procedures for greedy selection of faces and for face endpoint adjustment both make frequent use of subroutines for the ordering of a list of face records according to some key field value, and for analysing the nature of the overlap between two faces. These two aspects of the processing will now be described.

Overlap

As described above [5.3.4.3], it was considered appropriate to characterise the overlap between two faces by two real values giving the absolute length of the overlap region as a fraction of the two face lengths. Determining the overlap between two faces is less straightforward than might be expected. Given that the start and end points of two faces are available as chainlet numbers on the contour chaincode, say (L1,R1) and (L2,R2), there are many possible configurations to consider. The 'wraparound' causes further complications; one cannot assume, for example, that if R1<L2 then the two faces are disjoint.

In the implementation of overlap analysis an 'analogue' approach is used to simplify the testing: two arrays of length lcode are created and set to zero. The extents of the faces are then represented by setting the appropriate array elements to 1. Overlap is then found by logically ANDing these two binary patterns. If the number of 1s in the resultant array is zero then the two faces are disjoint. A single 1 implies that the faces have only one endpoint in common and this is taken as equivalent to disjointness.

It is conceivable that two faces could overlap at both ends giving two disjoint overlap regions, although this would almost certainly indicate an error in earlier processing stages. This eventuality is checked for and the program gives a warning message, but the situation was never encountered in practice.

The subroutine for overlap analysis returns two relative overlap figure and also an absolute measure in the form of the total length of the overlap region. In addition, the two arrays used to represent the faces are also made available for further analysis.

Sorting

Sorting the linked list of face records is implemented via an array. The sorting array is a linear array of two-element records; the array has as many elements as there are records to be sorted. The first field in the record holds a pointer to a face record in the linked list. The second record field holds the appropriate key value copied over from the face record which the first field pointer points to. This key value may be the face width or length, or whichever attribute value is required.

Any standard sorting algorithm may then be applied to the array, using the key values stored, with the modification that during the sorting process any rearrangement of the key fields is also applied to the pointer fields. At the end of the sort the key fields will be ordered and the pointers indicate the required order of the face records. The pointers may then be visited in turn and a new, ordered linked list constructed. In some situations it may not be necessary to reconstruct the linked list since the face records can simply be accessed via the sorting array.

This sorting procedure is straightforward to implement and convenient since the problem of ordering a linked structure has been translated into the standard problem of ordering an array of key values. In the C implementation the Shellsort algorithm [Sedgewick 1983, Ch 8] was used to sort the array.

5.3.7.3 Greedy selection

First the face records are ordered using the arcused field as key. This has the effect of bringing together the records for faces found on the same row of mmspace into continuous runs, with the face records for adjacent rows forming adjacent groups in the resulting sorting array. As outlined above [5.3.4.2], two processing passes are then made. On the first pass the faces for mmspace rows 1 and 2, 3 and 4,... are considered; on the second pass the faces for rows 0 and 1, 2 and 3, 4 and 5... When a limit has been set on the minimum face length of interest it may not be necessary to consider some rows of low index value, since these rows can only produce short faces. In practice a face length limit of 6 was used, and row 0 was not used in the greedy selection.

In each processing pass adjacent pairs of arcused values are considered in turn. The appropriate section of the sort array is found and the number of records noted. If the number of relevant records is non-zero then this subset of faces is re-ordered: a smaller sorting array 'subarray' is created with the face length value used as the key. These records are then sorted into increasing face length order.

The faces are then compared in pairs. The analysis is carried out via 'subarray', and the face records are referred to via the indices of this array. This allows the comparison to be performed using a nested for-loop:

```
for (j=size-1; j>0; j--)
     for (k=j-1; k>=0; k--)
          {
          Perform analysis for the two faces whose records are pointed
          to by subarray[j].face_ptr and subarray[k].face_ptr
          }
```

Faces which it is decided should be deleted are marked by setting the key field to -1. The first step in the analysis of the pair of face records is therefore to check that neither face has been marked for deletion. If the faces are both valid then the overlap between the two faces is found. As detailed above, the subroutine for overlap returns three values: the absolute overlap and two relative overlap values.

If the overlap is zero then no action is taken. If the shorter face is completely overlapped by the longer then the shorter face is marked for deletion by setting subarray[k].key to -1. If the shorter face is only partially overlapped then the overlap is eliminated from the shorter face by truncation. As is the calculation of the overlap, this is actually performed via a logical operation. The two arrays which represent the extent of the two faces on the contour may be termed f1 and f2, representing the longer and shorter faces respectively. The array f2 is then overwritten in the following operation:

f2[p] = f1[p] AND NOT(f2[p]).

f2 now represents the truncated shorter face. Actually, f2 is too short by one chainlet, since the two faces are allowed to have an endpoint in common. This single point is thus identified and restored to f2.

The new face endpoints are identified and the pseudolen, truelen, and aspectrat values are calculated for the truncated face. The length and aspect ratio are checked against the prevailing limits and if the face is too short or the aspect ratio is too large then the face is marked for deletion by setting subarray[k].key to -1. If the face is not to be deleted then subarray[k].key is set to the new true length. subarray is then re-ordered and the pairwise analysis re-started for this particular pair of mmspace rows.

When the processing is complete for this pair of rows of mmspace the subarray is copied back into the section of the original sorting array representing the list of all face records from which it has been taken. In the process of copying across the information, face records which have been marked for deletion are given the key value -1 in the larger sorting array. At the end of the pass, when every pair of rows has been considered, the face records are re-sorted. The presence of the negative keys for faces to be deleted means that the entries representing these faces will be pushed to the end of the sort array and will not interfere with the second processing pass.

The second pass is the same as the first except that the pairs of rows considered are 0 and 1, 2 and 3,... At the end of the second pass the linked list of face records is reconstructed according to the ordering given by the sorting array, excluding the array elements carrying negative key values, which represent deleted faces. The greedy selection procedure is then complete.

5.3.7.4 Endpoint adjustment

The greedy selection process removes partial overlap between faces found on adjacent rows of mmspace. The face endpoint adjustment process eliminates partial overlap between faces found at all resolutions.

The first step is to order the face records according to the widths of the faces. The width values used may be taken from the maxwid field in the face record. The value stored here is an approximation to the face width in the form of the upper limit of the threshold interval which found the face in the density slicing. This value has been found to be sufficiently accurate in practice for use in the ordering algorithm. Later implementations explicitly measured the face widths, using the same procedure as in the creation of the intermediate representation, to give values of sure accuracy.

The face records are sorted into increasing order of the corresponding face widths. Starting with the narrowest face, each face is compared with every face wider than itself. As in the greedy selection process the use of the sorting array allows this to be implemented in a simple nested loop ('face j' is used to denote the face whose record is pointed to by the pointer in the jth element of the sorting array):

```
for (j = 0; j < no_faces - 1; j++)
    {
    If sortarray[j].key ≠ -1 then find the direction of face j
    for (k = j+1; k < no_faces; k++)
        {
        If sortarray[k].key ≠ -1 then examine the overlap between face j
        and face k and process accordingly
        }
    }
```

The check on the keys is required to exclude faces which should be deleted. Since the directions of the faces will be used in deciding how to deal with partially overlapping faces, the direction of face j is calculated in the outer loop. Within the inner loop the overlap of faces j and k is calculated. As described above, the overlap analysis subroutine returns three values characterising the overlap. The overlap given as a fraction of the length of face j may be denoted 'over1', and given as a fraction of the length of face k may be denoted 'over2'.

If the total overlap is zero then no action need be taken. If a wider face is wholly or nearly covered by a narrower one then the wider face is deleted. This condition is realised as 'if (over2 > 0.8)'. The decision that a face should be deleted is recorded by setting the key value of the sorting array to -1. If the overlap of the wider face is less than 80% then the face survives this test. The characteristics of the overlap are then more closely examined. As discussed above [5.3.4.3], the decision to truncate or extend the wider face uses both information on the size of the relative overlap and the angular difference between the two faces. The direction of face k is therefore calculated from the end point data. The decision to extend is made if the angular separation, measured in degrees, is less than 45 times the fraction of the narrower face which is overlapped. Thus the closer the directions of the two faces, the more likely it is that the wider face will be extended to completely overlap the narrower.

The extension of face k is implemented by making use of the arrays f1 and f2 which were created in the overlap analysis and have been described above [5.3.7.2]. First the end of face k which has to be extended is identified. This is achieved by finding if f1 has value 0 or 1 at the positions corresponding to the end points of face k. If the beginning of the face has to be extended backwards this is implemented by copying over the start position information for face j from its face record and overwiting the corresponding face k record values. The face length and aspect ratio are then re-calculated and these record fields updated. If the end of face k has to be extended forwards this is achieved in a similar manner.

In the truncation of face k the first step is once more to identify which end has to be truncated, and the same method is used as for the extension procedure. If the beginning of face k has to be truncated forwards this is implemented by replacing the start position information in the face record with the end position information taken from the record of face j. The new length and aspect ratio for face k are then calculated. A check is included on whether the truncated face now has a length below the current face length limit or an aspect ratio above the current limit. If so, then the face is marked for deletion by setting the sortarray[k].key to -1.

If the face is truncated a final check must be made to see whether the resulting face is overlapped to more than 80% of its length by one of the narrower faces already considered. This is carried out by another small loop. If a narrow face is found to overlap the truncated face in this way then the truncated face is marked for deletion and this local iteration is stopped.

At the completion of the main nested loop all pairs of valid faces have been compared. The linked list of face records is then re-assembled from the sorting array, screening out any record which has been marked for deletion. The faces represented in this final linked list of records now form a strict hierarchy with no partial overlap, apart from the possible sharing of single end points.

At the end of the loop one more pass is made over the face records to update the face width and aspect ratio fields with accurate values calculated from the original chain code data, as in the creation of the intermediate representation. In making accurate measurements of face widths it was sometimes found that the estimates used in earlier stages of processing had slightly underestimated the values. Thus, sometimes it would be discovered that faces no longer satisfied the aspect ratio limit used. This was particularly unfortunate where only one level of structure had been found on the contour, since in these situations the only structure found would be deleted. In order to prevent this occurence the aspect ratio restriction was relaxed, and faces were not deleted if their aspect ratio values were less than 150% of the limit originally prescribed.

5.4 Testing and evaluation

Two aspects of the algorithm's performance will be evaluated: its effectiveness and its efficiency. In other words, one needs to check if the algorithm is doing the right thing and if it is doing the thing right. The efficiency of the algorithm will be considered separately below [5.4.4]. First the problem to be considered is how the effectiveness of the algorithm can be evaluated.

It would appear that algorithms of this type have not previously been evaluated in this way. Thus, for example, in the related works cited above [5.2.2], the authors evaluate their work by showing some examples and concluding that their algorithm "appears to be very effective" [Fischler & Bolles 1986], "yields good results in the cases tested" [Phillips & Rosenfeld 1987], "seemed to yield very good results" [Davis 1977a], or that "we believe the results to be satisfactory" [Bengtson & Eklundh 1989].

Given the difficulties involved in providing meaningful quantitative measures of effectiveness, such practice seems acceptable, provided that enough examples are tested and the examples are representative of the images to be analysed. Here, however, an attempt will be made to support such qualitative evaluations with some quantitative measures.

5.4.1 Metrics for performance

The aspect of performance considered here is the effectiveness of the algorithm; efficiency is dealt with separately below [5.4.4]. Petkovic [1989] has presented guidelines for establishing quantitative measures of the recognition accuracy of machine vision algorithms. His analysis is concerned with the performance of a recognition task such as automatic inspection, where an accept / reject decision is to be made about some component or object whose image is presented to the classifier. Petkovic's suggestions, in their original form, are inadequate for the problem of multi-resolution contour analysis considered in this chapter, and some modifications have therefore been made.

Petkovic presents two basic parameters which can be used to characterise the performance of a classification: the false negative rate (FN) and false positive rate (FP). The false negative rate, which may also be termed the miss rate or escape rate, is the ratio of the missed features to the total number of those features in the image. This may be reported as an overall rate or may be broken down into rates for different classes of features. The false positive rate serves to measure the extent to which the algorithm would report the presence of a feature which was not actually present in the image under test. Petkovic reports as suitable denominators the number of images or the number of features present.

Evaluating the performance of an algorithm designed for a complex task such as that addressed by the multi-resolution analysis algorithm is more problematic than evaluating some simple classification task. Firstly, the exact form of the ideal, 'right' answer may not be clear: there may not be an interpretation (i.e a set of faces) for the profile about which all informed observers would agree. Secondly, the algorithm may be in only partial agreement with the 'correct answer'. This leads to the problem of how, for example, one rates a suggested face which represents only 75% of the extent of the 'ideal face'. A further factor which must be taken into account is that the features to be found, i.e. faces on the contour, may be of widely different importance.

Thus, for example, the failure to identify a face of poor significance should not be considered as serious as the failure to find a highly significant face: some form of weighting may have to be introduced.

The performance analysis scheme which has been devised for the multiscale contour analysis algorithm aimed to produce quantitative measures akin to FN and FP. The nature of the problem being addressed, however, is such that a reliable measure of an algorithm's performance which is both objective and quantitative is probably impossible or impracticable to achieve. The scheme devised was intended to be relatively simple, yet still provide useful information of the required type.

5.4.1.1 Performance analysis method

For every image contour presented to the algorithm an estimate of the ideal answer is first made. This estimate is achieved by combining the analyses made by people who have been briefed on the problem. This process is considered in detail below [5.4.1.2]. The faces constituting the model solution are then divided into two classes: 'major' or 'significant' faces and 'minor' or 'insignificant' faces. This classification, again performed by hand, is the simplest way of providing some indication of the importance of the face. It would be possible to use a classification of three or more categories of significance; it may even be feasible to provide some quantitative significance value for each face, for example a real value in the range [0,1]. These strategies have not been attempted.

The possible faces on the contour which are found by the algorithm are then compared with those of the ideal solution. Major and minor faces are considered separately. In considering the errors of omission, i.e false negatives, three categories of misses are used. The categories are subjective. If the agreement between an 'ideal' face and suggested face is perfect or good then no miss is recorded. If no suggested face matches an ideal face, or if the agreement is poor, then a complete miss is recorded. An intermediate level of agreement is recorded as a partial miss, but if the suggested face extends too far beyond the ideal face then it is recorded as a complete miss.

The false positives are recorded in three similar categories. If a suggested face corresponds exactly to an ideal face or at least shows a good correspondence then no false positive is recorded. If a suggested face matches no ideal face or overlaps only slightly then it is recorded as a complete miss. An intermediate level of agreement is recorded as a partial miss.

This classification of the errors produced by the algorithm therefore depends on subjective judgements. Nevertheless it is considered that is does provide useful information on the performance of the algorithm. It also serves as a form of 'checklist', ensuring consideration of every 'true' face in the model solution and every face suggested by the algorithm. The provision of objective, quantitative performance metrics for such an algorithm can be seen to be a sizeable research problem, and one which has not been addressed in any of the published works on contour analysis cited in this chapter.

5.4.1.2 Establishing a reference solution

For each example profile analysed a reference solution was established against which the performance of the algorithm could be gauged. This was achieved by combining the analyses prepared by four volunteers, none of which had any experience of crystallography or geology.

The attached sheets show a series of two-dimensional outlines or "contours". On each figure please mark the sections of the contours which appear to you to be significantly straight. Please note the following points:

- Try not to be influenced by the symmetry of the figure - i.e. don't make any special effort to create a symmetric polygon.

- If sections of the contour do not appear to be significantly straight then ignore them. Do not feel obliged to extend your estimates of straight portions just in order to make them join up.

- Do not feel restricted to using one "scale" in your analysis. Some portions of contour may show alignments if examined at a small scale and a larger scale - see for example the figure below. If you think that significantly straight segments can be seen at more than one scale then please mark all of them.

- Since marking the contour may result in a cluttered diagram which is difficult to interpret, please mark the straight segments you perceive by using "rays" projecting from the contour at the segment endpoints. This technique is illustrated in the figure below, which shows an example contour section with rays added to mark the significantly straight segments found. In the figure the thick line represents the contour, the rays are marked by the broken lines and the thinner unbroken lines mark the extents of the perceived straight sections. The offset of the unbroken lines from the contour has no significance.

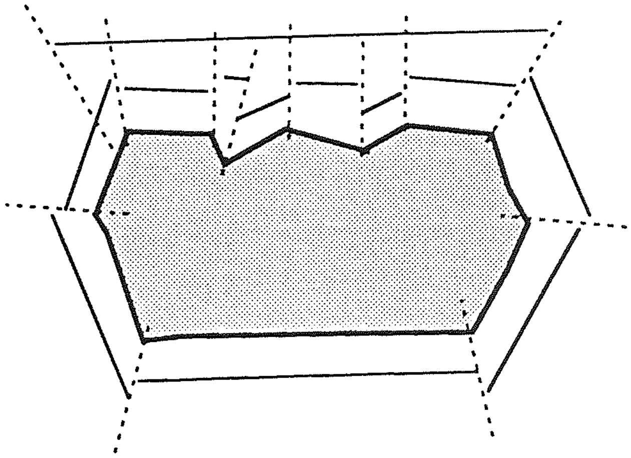


Figure 5.36 Instructions for preparing hand analyses

The briefing notes given to the volunteers are reproduced as figure 5.36. Volunteers were also provided with a card with slots of different widths for viewing sections of the contour in isolation; some did not make use of this.

The various solutions provided were then combined. This involved trying to find a compromise, 'consensus' solution where each face corresponded to at least one of the suggested faces, and there was a face with which each suggested face could correspond. If a proposed face was supported by only one interpretation, however, it could be dropped, but some subjective judgment was used. There was found to be a broad agreement between the different analyses, which simplified this process. It is considered that the final consensus solutions do represent the linear and near-linear structure perceived on the contours fairly well, but by the nature of the task in some cases there must always remain some room for argument.

5.4.1.3 Example performance analysis

As an illustration of the performance analysis method, the analysis of the 'pentagonal' contour (figure 5.13(a)) will be considered. This profile was not one of the test examples analysed by volunteers: the representative, 'ideal' solution was in this case prepared by hand. Figure 5.37(a) shows the 'faces' of the consensus solution. Six of these faces have been classed as major: the five faces which define the basic pentagonal shape and the pefectly straight subface of the oblique lower left face. Twenty three minor faces have also been suggested. Thus the maximum possible numbers of false negatives (i.e. misses) are 6 major faces and 23 minor faces. It may be noted that these faces do not form a true face tree because of some partial overlap among subfaces at the left end of the upper major face.

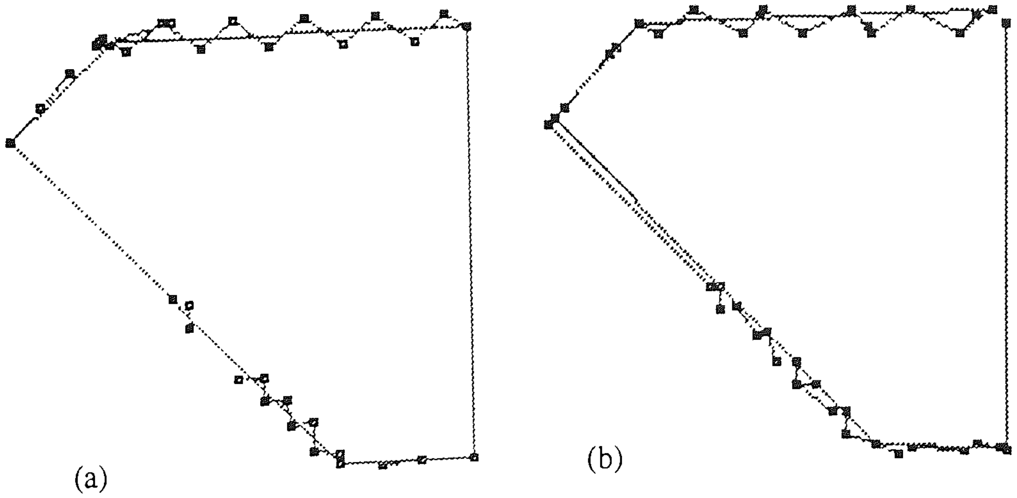


Figure 5.37 The consensus solution and multiscale analysis of the 'pentagon' test contour

Figure 5.37(b) shows the results of the multiscale analysis; the creation of this figure (fig.5.25(c)) has been described above [5.3.4.5]. This analysis has suggested twenty nine faces, of which six can be classed as major. Thus the maximum possible numbers of false positives are 6 major faces and 23 minor faces. The six major faces found correspond to the six major faces of the consensus solution. Five minor faces present in the consensus solution correspond only partially to faces suggested by the multiscale analysis, and one has no counterpart. Conversely, the multiscale analysis has suggested five minor faces which correspond only partially to faces found in the consensus solution, and one which has no counterpart.

Table 5.1 summarises the performance analysis for this example. Table 5.1(a) records the 'false negatives' or misses of the algorithm; 5.1(b) records the 'false positives' or exaggerations of the algorithm.

(a) FN error summary:

class	Major	Minor
Partial miss	0	5
Full miss	0	1
Max possible error count	6	23

(b) FP error summary:

class	Major	Minor
Partial miss	0	5
Full miss	0	1
Max possible error count	6	23

Table 5.1 Performance summary for 'pentagon' example

The tables clearly show how the algorithm has been effective in identifying the major faces on the contour and has identified most of the minor faces while at the same time producing few spurious suggestions of linear structure.

5.4.2 Range of tests and results

The multi-scale contour analysis algorithm has been tested on a series of real and synthetic images. The version of the algorithm tested is the one which handles perfectly straight faces as a special case, excluding them from the greedy selection [5.3.4.2]. The analysis contains the final filtering pass which aims to remove 'step' faces introduced by digitisation and other unnecessary minor faces [5.3.4.4].

Nine real data examples were used. These were derived from published thin section photomicrographs. The topaz and lamprophyllite crystal profiles have already appeared (figures 5.17(a) and 5.21); the other seven crystal profiles were, like the lamprophyllite example, produced by hand digitisation. The two synthetic images have also been used. These are versions of test images which have been used in published work on contour analysis.

The test contour examples are presented together with the results of three different types of analysis. For each profile the 'model analysis', or 'consensus solution' is briefly described. The results of the multi-scale analysis algorithm are then presented, for both the slower and faster arc length progressions. Finally, the result of applying Lowe's striptree-based algorithm for contour analysis [5.2.2.4] is shown for comparison.

5.4.2.1 Topaz

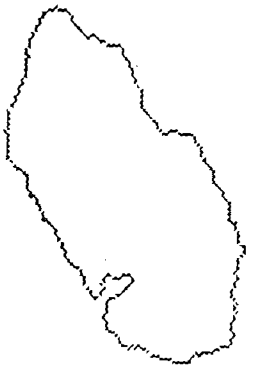


Figure 5.38 Topaz crystal profile

Figure 5.38 reproduces figure 5.17(a); this is the topaz crystal test profile. The consensus solution for this figure found eight major faces and thirteen minor faces, with up to three levels of nesting among the faces and subfaces on the upper right of the figure.

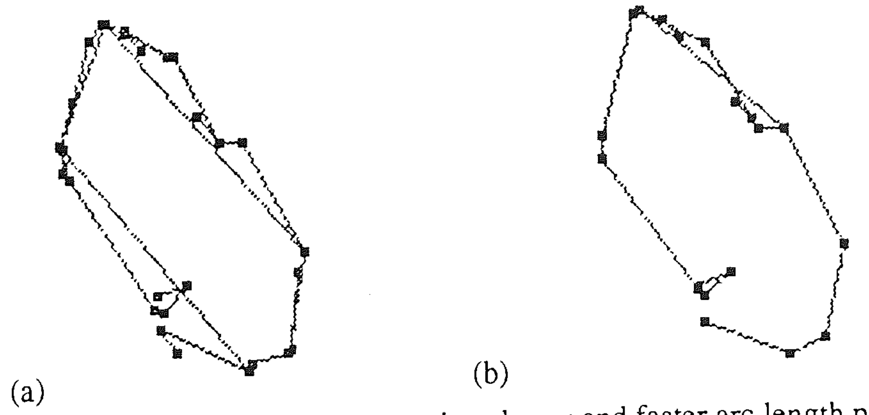


Figure 5.39 Analysis of test profile using slower and faster arc length progressions

Figure 5.39(a) shows the results of the multiscale analysis using the slower arc length progression. This is in broad agreement with the consensus solution; only three of the suggested faces were not picked out by the algorithm. One of these, a subface of the oblique lower left face, was found by the density slicing but was removed during the greedy selection when it was rejected in preference to the containing face, since the two face markers were found on adjacent rows of mmspace.

Figure 5.39(b) shows the results of the multiscale analysis using the faster arc length progression. As noted above [5.3.3.5], this analysis has not picked up the two long oblique faces present in figure 5.39(a), but has suggested a face present in the consensus solution which the previous analysis did not find.

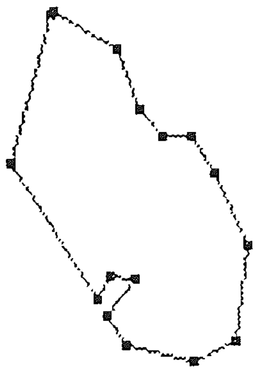


Figure 5.40 Analysis of test profile using Lowe's algorithm

Figure 5.40 shows the result of applying Lowe's algorithm. As previously noted [5.2.2.4], the face on the right hand side has been found as two subfaces which, since they lie in separate subtrees of the strip tree, cannot be merged by Lowe's algorithm.

Table 5.2 summarises the performance of the multiscale analysis algorithm applied to the topaz crystal profile.

Example: topaz

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	0	0	0
Full miss	0	3	0	4
Max possible error count	8	13	8	14

Table 5.2 Performance summary for topaz example

5.4.2.2 Lamprophyllite

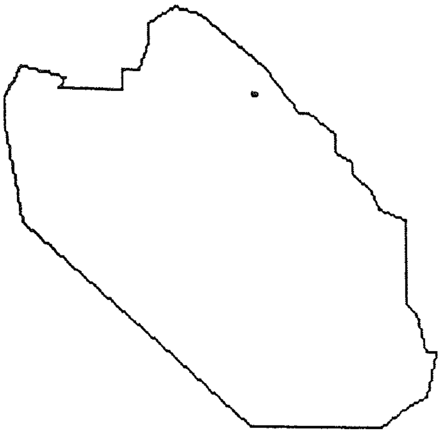


Figure 5.41 Lamprophyllite crystal profile

Figure 5.41 reproduces figure 5.21: the lamprophyllite crystal test profile [MacKenzie & Guilford 1980, example 53]. The consensus analysis for this profile found eleven major faces and twenty minor faces. Figure 5.42(a) shows the results of the multiscale analysis using the slower arc length progression. One of the major faces, the small face on the left of the embayment, was only partially picked up in this analysis. In addition, four minor subfaces on the noisy long upper face were not found, some of them being lost in the final filtering pass. The analysis produced no faces which were not at least partially represented in the consensus solution.

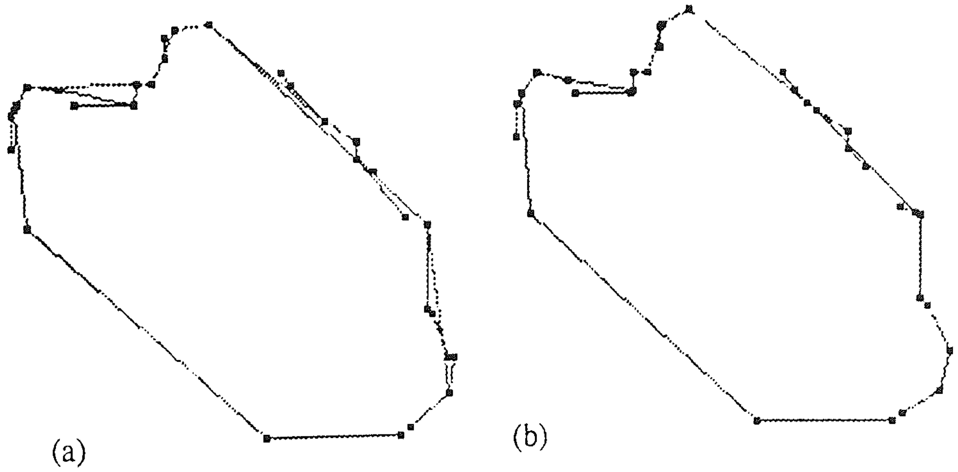


Figure 5.42 Analysis of test profile using slower and faster arc length progressions

Figure 5.42(b) shows the results of the multiscale analysis using the faster arc length progression. This analysis has failed to pick up some of the more irregular faces, notibly the long upper face and the oblique face on the right hand side. Figure 5.43 shows the results of applying Lowe's algorithm, which has produced a good description of the less noisy faces on the profile.

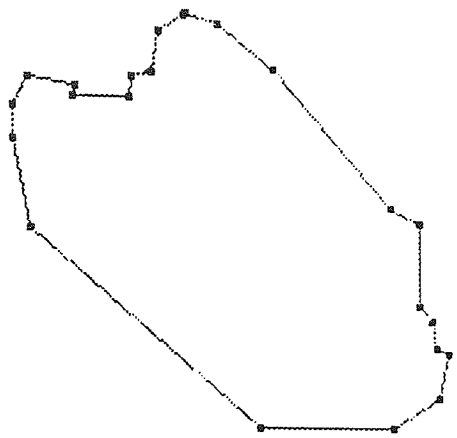


Figure 5.43 Analysis of profile using Lowe's algorithm

Table 5.3 summarises the performance of the multiscale analysis algorithm applied to the lamprophyllite crystal profile.

Example: lamprophyllite

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	1	1	0	1
Full miss	0	4	0	0
Max possible error count	11	20	10	16

Table 5.3 Performance summary for lamprophyllite example

5.4.2.3 Olivine 1

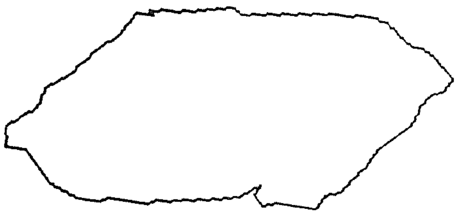


Figure 5.44 First olivine crystal profile

The test profile of figure 5.44 is that of an olivine crystal, hand-digitised from a published micrograph [MacKenzie & Guilford 1980, example 1]. The consensus analysis for this profile found eight major faces and twenty minor. Figure 5.45(a) shows the results of the multiscale analysis using the slower arc length progression. This analysis has found all the major faces of the consensus solution: six faces defining a basic hexagonal profile with two important subfaces on the lower face separated by the small embayment.

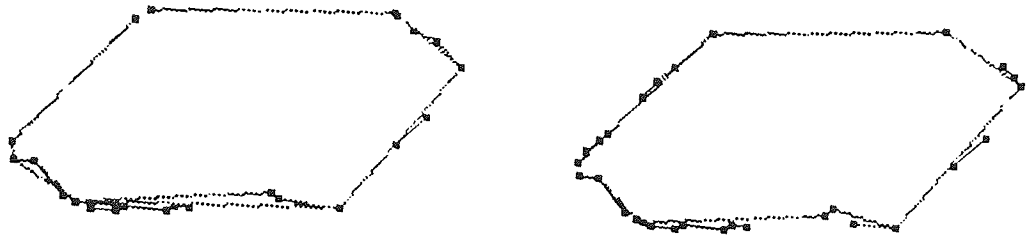


Figure 5.45 Analysis of test profile using slower and faster arc length progressions

The analysis failed to pick up nine minor faces present in the consensus solution. The majority of these were subfaces of the four relatively straight major faces; approximately half of these subfaces had been found by the algorithm but were removed in the final filter pass. Figure 5.45(b) shows the results of the multiscale analysis using the faster arc length progression. This analysis has not found the two relatively noisy major faces on the base and bottom left of the profile, but has found more detail on the upper faces.

Figure 5.46 shows the results produced by Lowe's algorithm; the basic hexagonal form has not been found by this analysis.

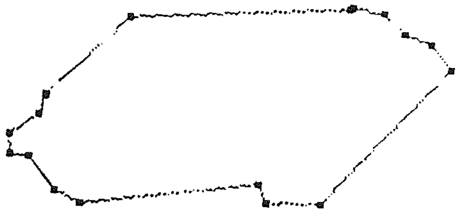


Figure 5.46 Analysis of profile using Lowe's algorithm

Table 5.4 summarises the performance of the multiscale analysis algorithm applied to the first olivine crystal profile.

Example: olivine 1

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	0	0	1
Full miss	0	9	0	0
Max possible error count	8	20	8	11

Table 5.4 Performance summary for first olivine example

5.4.2.4 Allanite

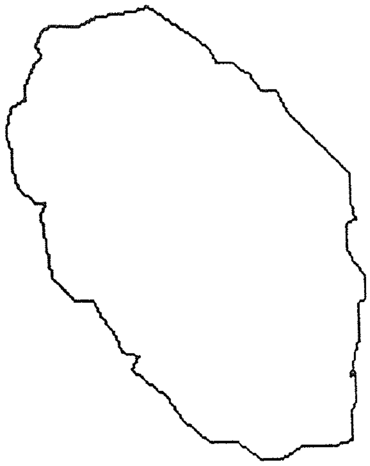


Figure 5.47 Allanite crystal profile

The test profile of figure 5.47 is that of an allanite (orthite) crystal, hand-digitised from a published micrograph [MacKenzie & Guilford 1980, example 25]. This figure shows linear and near-linear structure at a range of scales. The consensus analysis found eleven major faces and thirty minor. The right hand side of the figure was seen as two long irregular faces, each with several subfaces. The lower side of the figure was interpreted as one irregular face, and also as five subfaces.

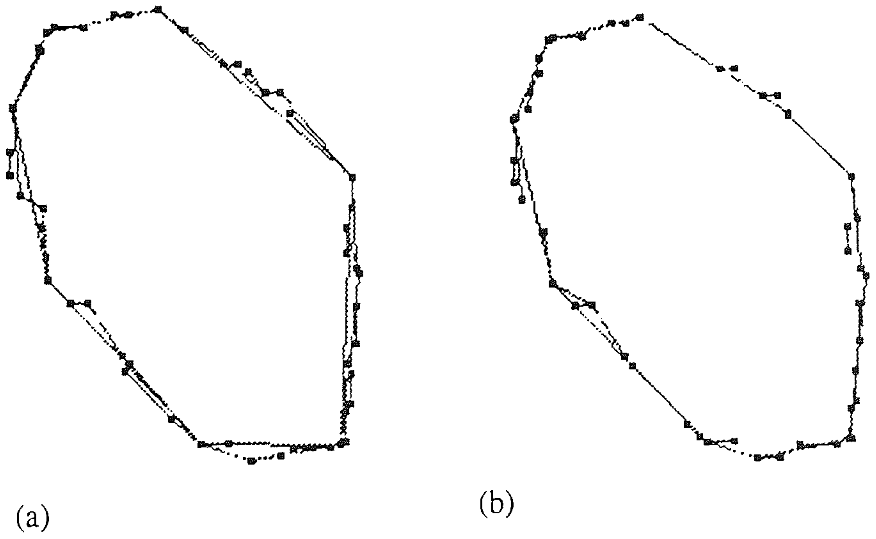


Figure 5.48 Analysis of test profile using slower and faster arc length progressions

Figure 5.48(a) shows the results of the multiscale analysis using the slower arc length progression. This analysis has suggested forty five faces, ten of which could be classed as major. The analysis only picked up part of the relatively straight section at the top of the upper right oblique face; otherwise the major faces were found correctly. Minor faces were less well analysed, with nine minor faces lost and twelve suggestions made for faces not present in the consensus solution. The majority of these errors, however, are of little weight since they represent very small faces.

Some of the 'false positive' faces suggested are digitisation step faces which have not been removed in the final filter pass. The loss of two oblique subfaces on the lower face has been traced back to the greedy algorithm: the correct faces were found by density slicing but were rejected when compared with the longer oblique faces whichare still present in the analysis. This loss of useful faces in the greedy selection process is similar to that found in the topaz example.

Figure 5.48(b) shows the results of the multiscale analysis using the faster arc length progression. This analysis is less satisfactory than figure 5.48(a). Four of the long, irregular faces have not been found: on the upper and lower right hand side, the lower face and the lower left hand side, where only two subfaces have been found. Figure 5.49 shows the results produced by Lowe's algorithm. This has found the less noisy faces in preference to the longer, more irregular possibilities.

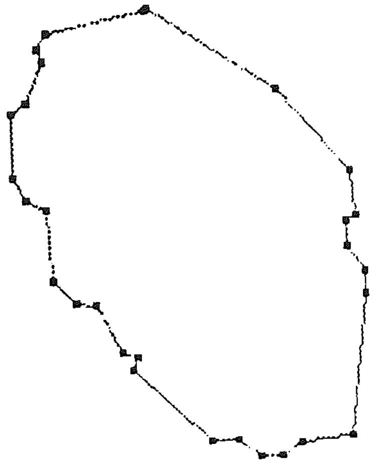


Figure 5.49 Analysis of allanite profile using Lowe's algorithm

Table 5.5 summarises the performance of the multiscale analysis algorithm applied to the allanite crystal profile.

Example: allanite

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	1	1	0	4
Full miss	0	9	0	12
Max possible error count	11	30	10	35

Table 5.5 Performance summary for allanite example

5.4.2.5 Augite 1

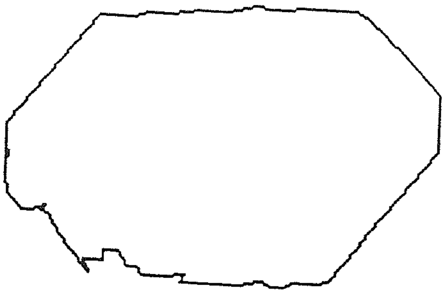


Figure 5.50 First version of first augite crystal profile

The test profile of figure 5.50 is that of an augite crystal, hand-digitised from a published micrograph [Cox *et al.* 1974, p139]. The contour shows a basic octagonal form disrupted by irregularities on the top left side. The consensus analysis found eight major faces and sixteen minor subfaces within the noisier vertical sides. Figure 5.51(a) shows the results of the multiscale analysis using the slower arc length progression. The principal faces have been correctly identified, as have most of the subfaces. The analysis also produced few suggestions for faces not present in the consensus solution: only four minor faces.

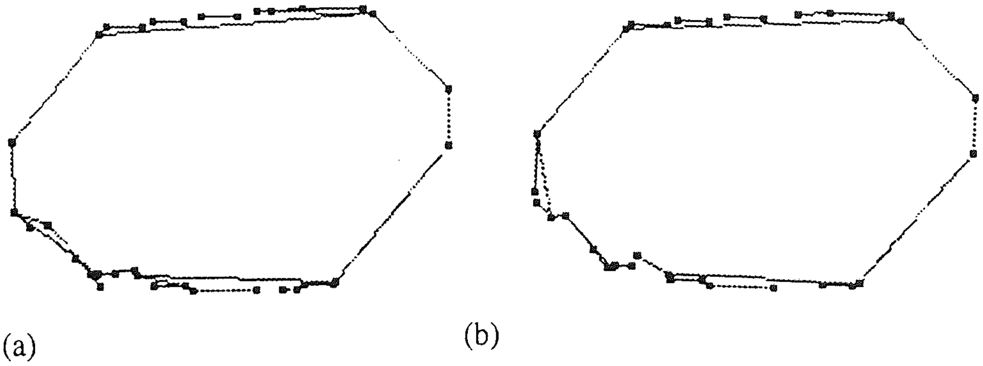


Figure 5.51 Analysis of test profile using slower and faster arc length progressions

As in the topaz example a 'corner has been cut' by the algorithm, where one of the two limbs on a corner has been replaced in the greedy selection by a wider subface spanning both limbs. In this case this has occurred near the left end of the top face. This problem did not occur in figure 5.51(b): the result of the multiscale analysis using the faster arc length progression. This analysis is very like figure 5.51(a), but with the notable loss of the suggestion of an oblique face at the upper left.

Figure 5.52 shows the result of applying Lowe's algorithm to the contour.

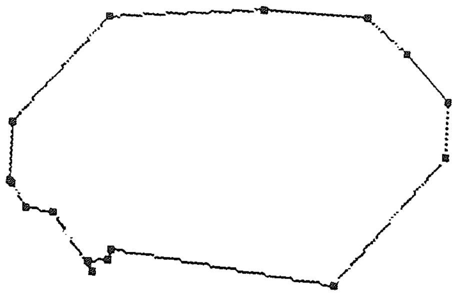


Figure 5.52 Analysis of test profile using Lowe's algorithm

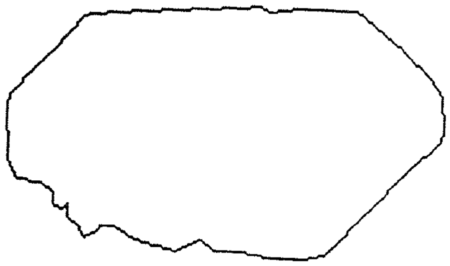
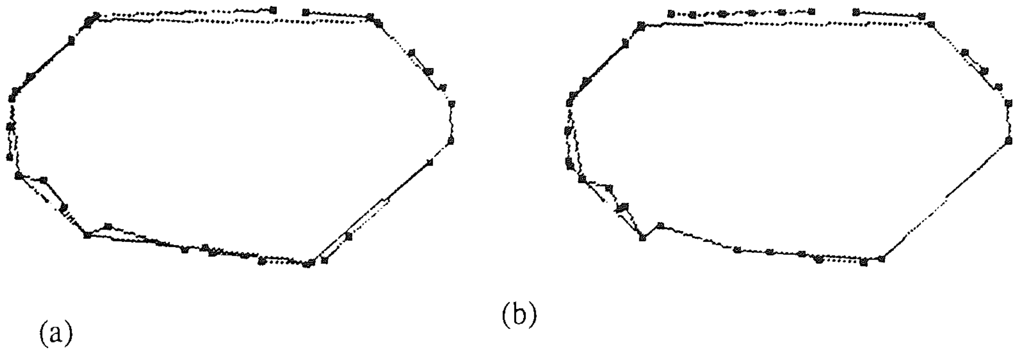


Figure 5.53 Second version of first augite crystal profile

Figure 5.53 shows an alternative digitisation of the same augite crystal profile. In this case the consensus analysis found nine major faces and twelve minor. Figure 5.54(a) shows the results of the multiscale analysis using the slower arc length progression. The analysis is relatively successful, missing only two minor faces. Eight faces not present in the consensus solution were suggested, however, mainly as subfaces of the relatively straight faces. The final filter pass might have been expected to remove some of these subfaces, but a condition for removal is that the subface should be less than 15% the length of the containing face, and this condition is not met here.



(a)

(b)

Figure 5.54 Analysis of test profile using slower and faster arc length progressions

Figure 5.54(b) shows the results of the multiscale analysis using the faster arc length progression. The analysis is similar to figure 5.54(a), but with the loss of the long irregular face on the left of the figure, and some additional, unnecessary subfaces on the right side. Figure 5.55 shows the result of applying Lowe's algorithm. As with figure 5.53, this analysis has unexpectedly split some of the better faces perceptible on the contour.

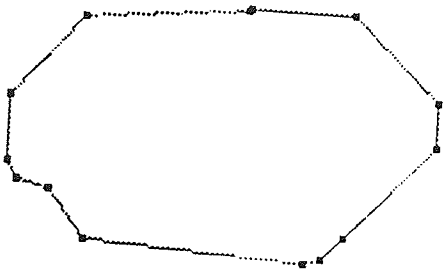


Figure 5.55 Analysis of second version of first augite profile using Lowe's algorithm

Table 5.6 summarises the performance of the multiscale analysis algorithm applied to the two versions of the first augite crystal profile.

Example: augite 1

Error summary:

(a) 1st version	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	0	0	0
Full miss	0	2	0	4
Max possible error count	8	16	8	18

(b) 2nd version	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	1	0	1
Full miss	0	2	0	8
Max possible error count	9	12	9	18

Table 5.6 Performance summary for two versions of first augite example

5.4.2.6 Augite 2

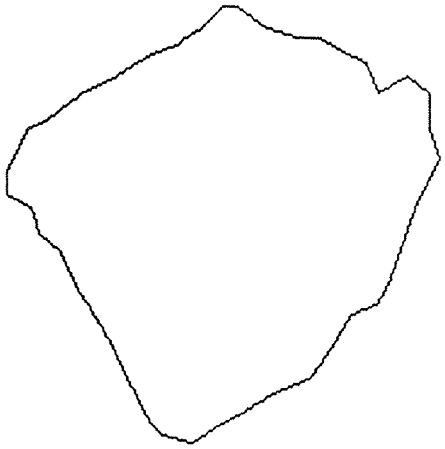


Figure 5.56 Second augite crystal profile

The test profile of figure 5.56 is that of an augite crystal, hand-digitised from a published micrograph [Cox *et al.* 1974, p139]. Inaccuracy in the digitisation in this case has concealed to some extent the euhedral quality of the original. The figure is roughly pentagonal, but there is some ambiguity about the exact limits of the major faces. The consensus analysis found seven major faces and twenty one minor.

Figure 5.57(a) shows the results of the multiscale analysis using the slower arc length progression. This analysis has found all the major faces, but failed to identify five minor subfaces. Probably the clearest loss concerns two minor faces to the right of the small embayment on the noisy upper right face. The loss of these faces has once more been traced to the greedy algorithm. The faces were correctly found by the density slicing, together with one oblique face spanning them both. The presence of this longer face led to the removal of the two subfaces in the greedy selection, and the longer face was then removed in the endpoint adjustment stage where a more accurate measurement found the face to lie outside the limit set on aspect ratios.

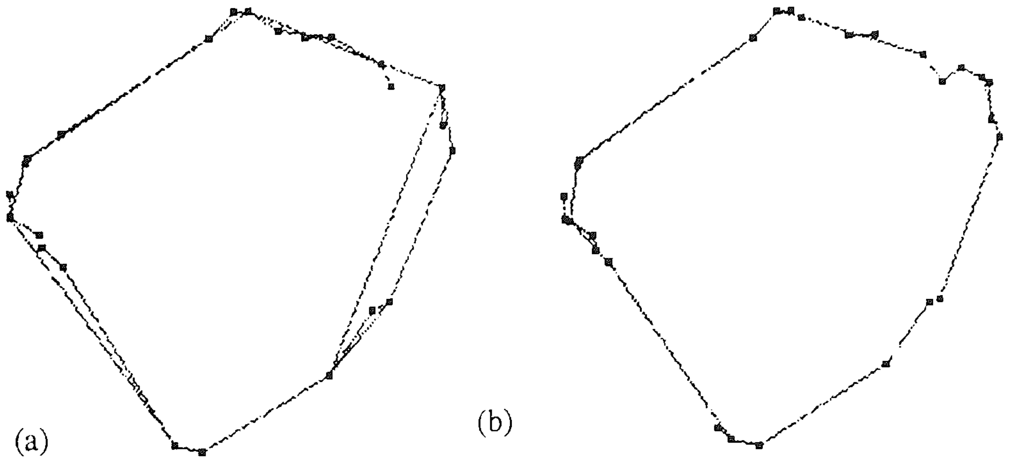


Figure 5.57 Analysis of test profile using slower and faster arc length progressions

Figure 5.57(b) shows the results of the multiscale analysis using the faster arc length progression. This analysis has lost some of the more irregular faces present in figure 5.57(a). It may be noted that the two subfaces lost in the previous analysis have been found correctly here, since no face was in a position to compete against them in the greedy selection.

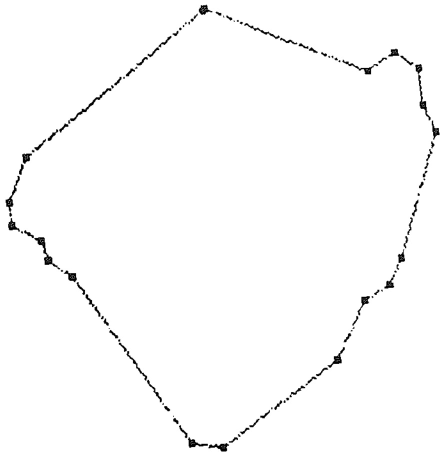


Figure 5.58 Analysis of profile using Lowe's algorithm

Figure 5.58 shows the result of applying Lowe's algorithm. The analysis is similar to figure 5.57(b), but presents a poorer analysis of the upper right long oblique face. Table 5.7 summarises the performance of the multiscale analysis algorithm applied to the second augite crystal profile.

Example: augite 2

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	1	0	1
Full miss	0	4	0	3
Max possible error count	7	21	7	19

Table 5.7 Performance summary for second augite example

5.4.2.7 Nosean

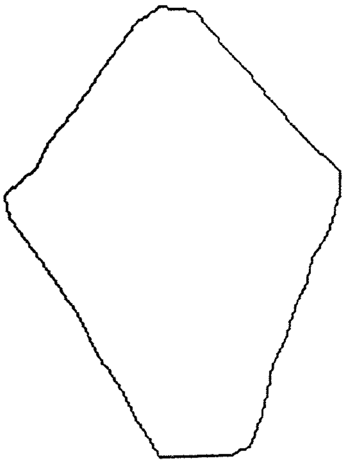


Figure 5.59 Nosean crystal profile

The test profile of figure 5.59 is that of a nosean crystal, hand-digitised from a published micrograph [MacKenzie & Guilford 1980, example 80]. This profile shows a hexagonal form, although the exact extent of the faces is sometimes not clear because of slight changes in direction near the end points. The consensus analysis for this profile found six faces defining the basic hexagon, five of which faces were classed as major, and also picked up another seven possible minor subfaces.

Figure 5.60(a) shows the results of the multiscale analysis using the slower arc length progression. The six faces of the hexagon have been found, and two interpretations of the lower face have been proposed, which corresponds to the consensus solution. Two subsegments of the lower right oblique face have been suggested as possible faces, but only the smaller of these is found in the consensus solution. The consensus analysis suggests subfaces within the upper left face, which the multiscale analysis did not find, although one was removed only in the final filter pass.

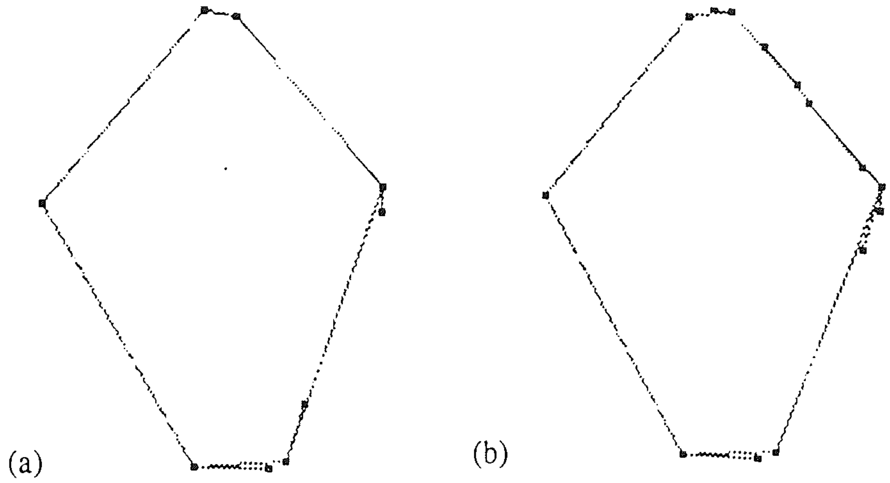


Figure 5.60 Analysis of test profile using slower and faster arc length progressions

Figure 5.60(b) shows the results of the multiscale analysis using the faster arc length progression. In its main features this result is clearly similar to figure 5.60(a), with the loss of no major faces. This analysis has suggested subfaces within the upper, upper right and lower right faces, but none of these were found in the consensus solution.

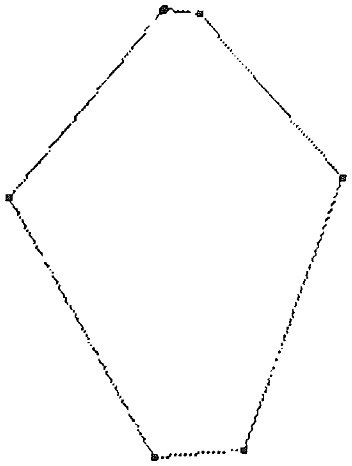


Figure 5.61 Analysis of test profile using Lowe's algorithm

Figure 5.61 shows the result produced by Lowe's algorithm, where the six sides of the basic hexagon have been found. Table 5.8 summarises the performance of the multiscale analysis algorithm applied to the nosean crystal profile.

Example: nosean

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	0	0	0
Full miss	0	5	0	1
Max possible error count	5	8	5	4

Table 5.8 Performance summary for nosean example

5.4.2.8 Olivine 2

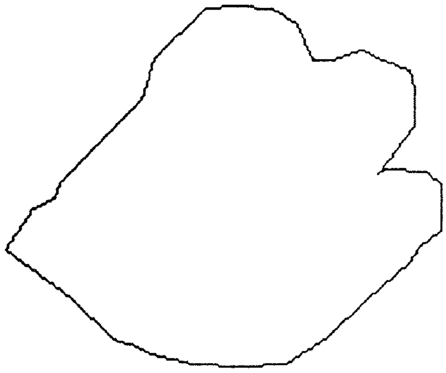


Figure 5.62 Second olivine crystal profile

The test profile of figure 5.62 is that of an olivine crystal, hand-digitised from a published micrograph [Mackenzie *et al.* 1982, example 1982]. This is a subhedral olivine, which in the original micrograph shows curved, amorphous contour segments. The digitisation process has, however, introduced noise and poorly approximates some of the curves. The figure shows two clear subparallel faces, with the one on the upper left, according to the consensus analysis, showing five possible subfaces. The curved contour segment connecting these two sides was viewed as two likely faces, each with two possible subfaces.

Figure 5.63(a) shows the results of the multiscale analysis using the slower arc length progression. The analysis did not find one of the expected faces on the base of the figure, suggesting only the two constituent subfaces. The analysis also picked up at least seven step faces created as side-effects of the digitisation which were too long to be removed in the final filter pass, i.e. they represented more than 15% of their containing faces. The analysis otherwise found all the important faces, and also made the useful suggestion of one irregular face across the top of the figure.

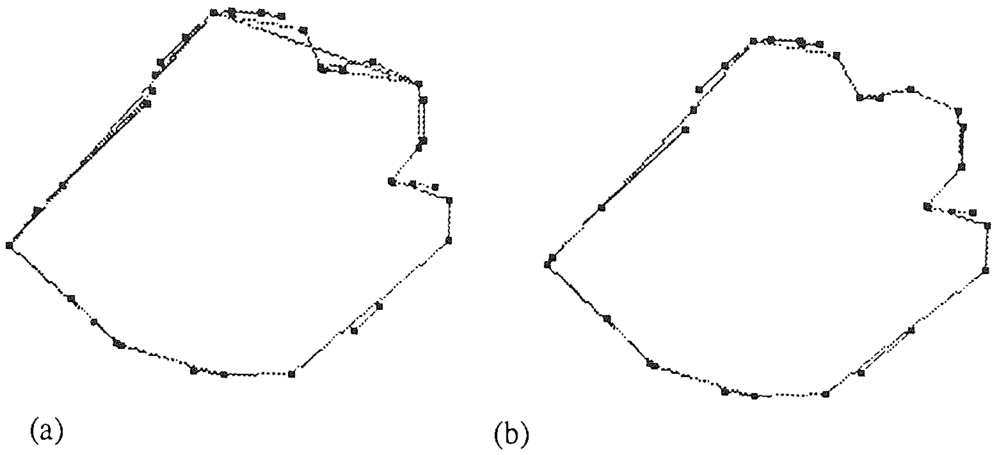


Figure 5.63 Analysis of test profile using slower and faster arc length progressions

Figure 5.63(b) shows the results of the multiscale analysis using the faster arc length progression. The results are similar, but with the loss of the upper irregular face. Figure 5.64 shows the results of applying Lowe's algorithm.

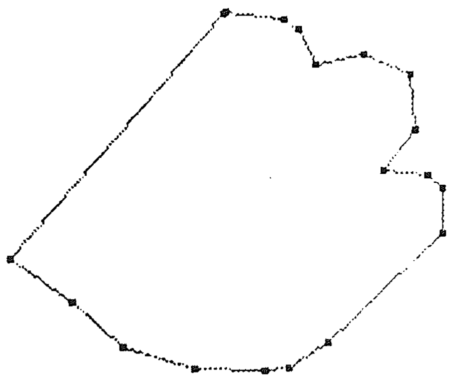


Figure 5.64 Analysis of test profile using Lowe's algorithm

Table 5.9 summarises the performance of the multiscale analysis algorithm applied to the second olivine crystal profile.

Example: olivine 2

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	3	0	1
Full miss	1	5	0	10
Max possible error count	9	20	10	22

Table 5.9 Performance summary for second olivine example

5.4.2.9 Arrow

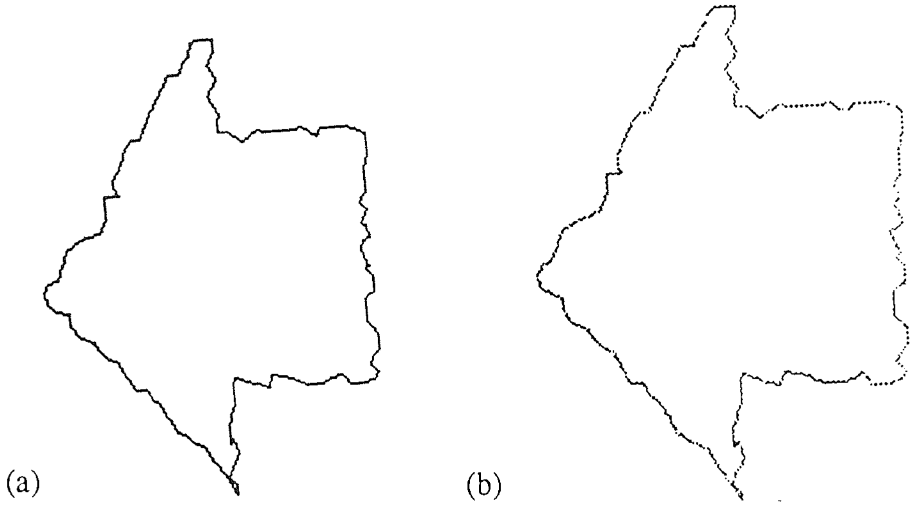


Figure 5.65 Arrow test profile at two scales

The test profiles of figures 5.65(a) and (b) are versions, at different scales, of a figure used by Bengtson *et al.* [1986, fig.8.5]. Neither figure is identical to the original since the figure was again hand-digitised, with the introduction of some noise. The original was digitised on a 1000 X 1000 grid; figure 5.65(a) was scaled to fit a 180 X 180 grid and figure 5.65(b) was scaled to fit a 200 X 200 grid. The consensus analysis for this figure showed seven major faces and twenty seven minor faces.

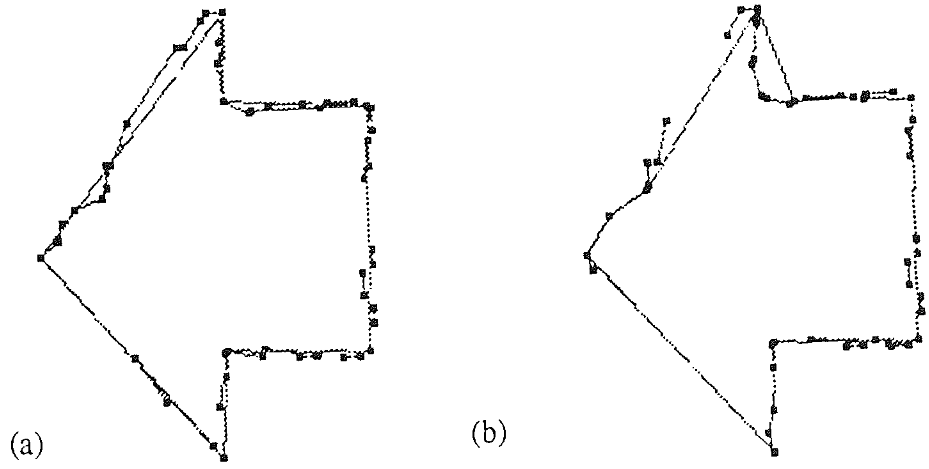


Figure 5.66 Analysis of test profile using slower and faster arc length progressions

Figure 5.66(a) shows the results of the multiscale analysis of figure 5.65(a) using the slower arc length progression. All major faces have been successfully identified, and few minor faces have been missed. Figure 5.66(b) shows the results of the multiscale analysis using the faster arc length progression. This analysis has lost three of the major faces. It is not clear from the figure if the other major faces have been correctly identified, but an examination of face markers in mmspace confirms that this is so. Figure 5.67 shows the result of applying Lowe's algorithm.

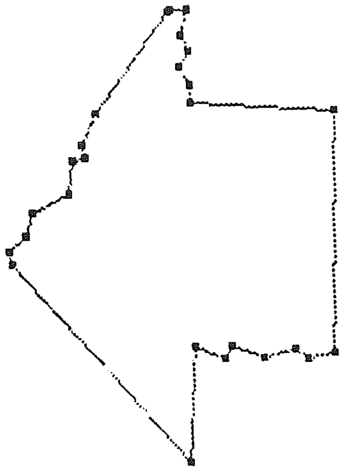


Figure 5.67 Analysis of smaller arrow profile using Lowe's algorithm

Table 5.10 summarises the performance of the multiscale analysis algorithm applied to the smaller 'arrow' test profile.

Example: arrow

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	0	1	0	3
Full miss	0	3	0	7
Max possible error count	7	27	7	33

Table 5.10 Performance summary for 'arrow' example

Figure 5.68(a) shows the results of the multiscale analysis, using the slower arc length progression, applied to figure 5.65(b). Figure 5.65(b) shows the results of the analysis using the faster arc length progression. Results can be seen to be similar to the previous analyses, with a similar loss of some of the noisier major faces when the faster arc length progression is used.

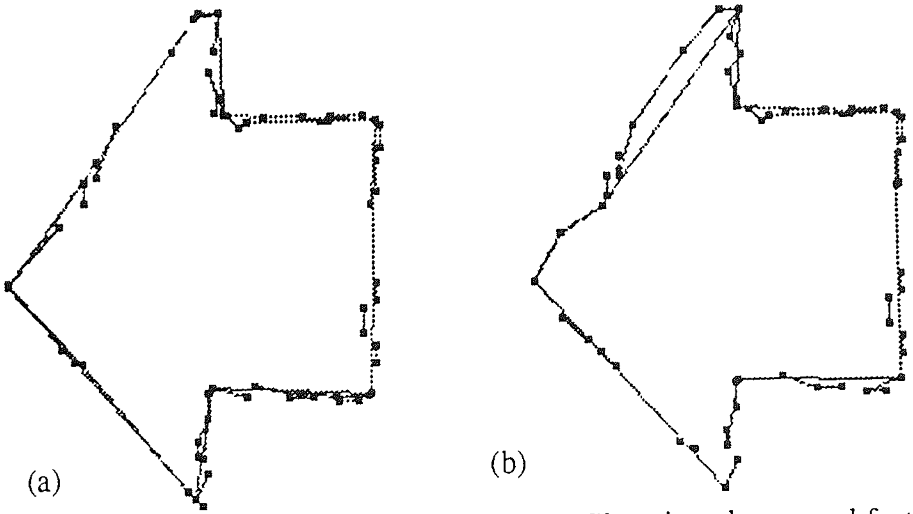


Figure 5.68 Analysis of larger 'arrow' test profile using slower and faster arc length progressions

5.4.2.10 Heptagon

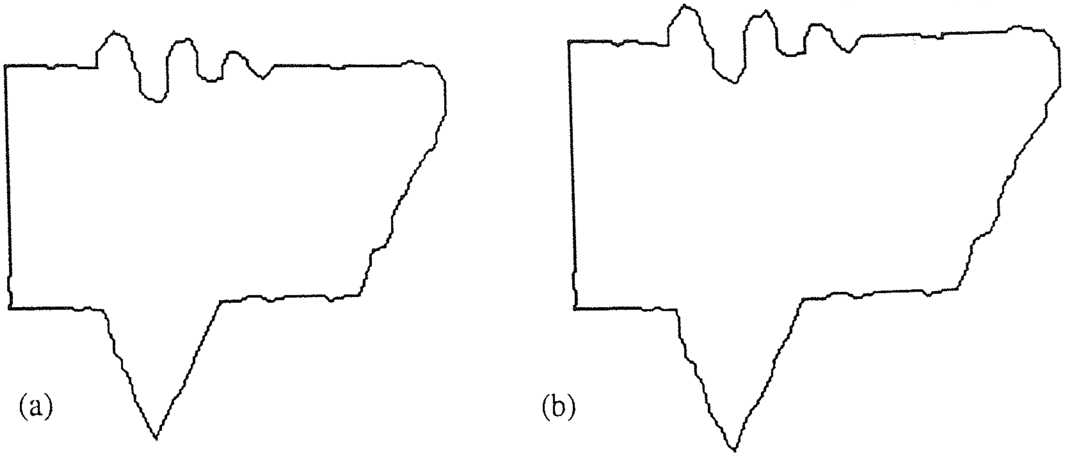


Figure 5.69 Heptagon test profile at two scales

The profiles of figures 5.69(a) and (b) are versions, at different scales, of a noisy irregular, roughly heptagonal test figure used by Bengtson *et al.* [1986, fig.8.2] and also by Fischler and Bolles [1986]. As with figure 5.65, hand-digitisation has introduced noise. The original was digitised on a 1000 X 1000 grid; figure 5.69(a) was scaled to fit a 180 X 180 grid and figure 5.69(b) was scaled to fit a 200 X 200 grid.

The consensus analysis of this profile found nine major faces and nineteen minor, ten of these on the noisy upper face. Two interpretations of the left hand side were suggested: one face extending the whole length of the side and one covering only the perfectly straight portion.

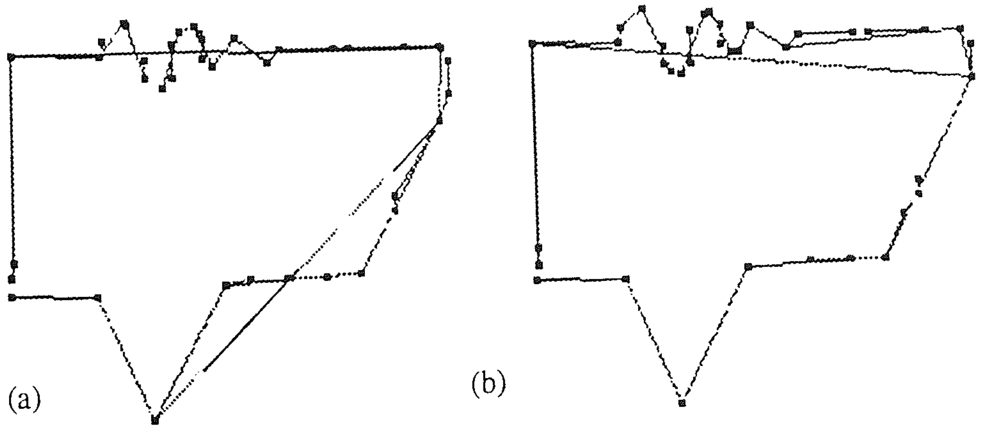


Figure 5.70 Analysis of smaller 'heptagon' test profile using slower and faster arc length progressions

Figure 5.70(a) shows the results of the multiscale analysis of figure 5.69(a) using the slower arc length progression. All major faces have been found, although only the perfectly straight left hand face was found. Both options for this side were found during the analysis, but the longer of the pair was removed during endpoint adjustment because of the rule which deletes faces with good subfaces which represent at least 80% of their length.

This analysis has also suggested a long, very irregular oblique face on the lower right of the profile, with three major faces as its subfaces. This face was not found in the consensus solution, but it does seem a reasonable suggestion when it is borne in mind that the analysis at this stage cannot take into account the nature of the rest of the contour. The analysis has suggested seven other faces not found in the consensus solution, mainly small faces associated with the noisy segment of the upper side.

Figure 5.70(b) shows the results of the multiscale analysis using the faster arc length progression. The analysis is similar, with the clearest differences being the loss of the very irregular left hand face and a change in the right hand end of the noisy upper face. Figure 5.71 shows the result of applying Lowe's algorithm.

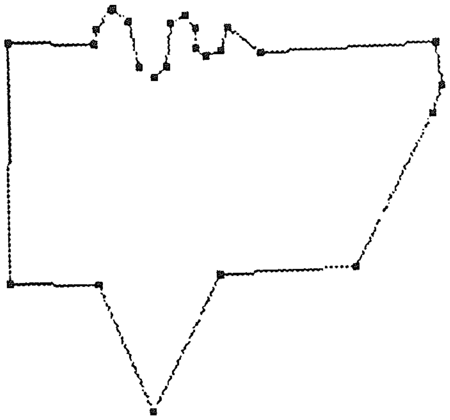


Figure 5.71 Analysis of smaller 'heptagon' profile using Lowe's algorithm

Table 5.11 summarises the performance of the multiscale analysis algorithm applied to the smaller 'heptagon' test crystal profile.

Example: heptagon

Error summary:	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	1	2	0	1
Full miss	0	5	0	8
Max possible error count	9	19	9	23

Table 5.11 Performance summary for 'heptagon' example

Figure 5.72(a) shows the results of the multiscale analysis, using the slower arc length progression, applied to figure 5.69(b). Figure 5.72(b) shows the results of the analysis using the faster arc length progression. The analyses can be seen to be broadly similar to those of figure 5.70.

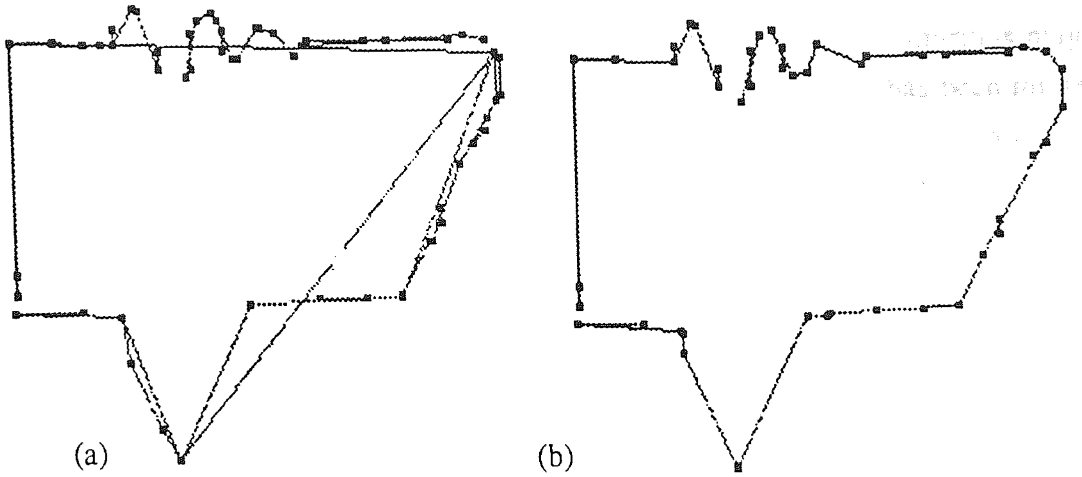


Figure 5.72 Analysis of larger 'heptagon' test profile using slower and faster arc length progressions

5.4.3 Analysis of results

Table 5.12 combines the results of the eight real crystal profile analyses, summing the values for each field in the table. In the case of the first augite example only the analysis of the first digitisation was used. Table 5.13 adds to table 5.12 the results for the two synthetic profile analyses. These tables therefore indicate the effectiveness of the algorithm over the above test examples.

	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	1	6	0	8
Full miss	1	41	0	34
Max possible error count	68	148	66	139

Table 5.12 Performance summary for all real data examples

	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	2	9	0	12
Full miss	1	49	0	49
Max possible error count	84	194	82	195

Table 5.13 Performance summary for all test examples

These tables suggest that the multiscale analysis algorithm is effective in identifying the major faces (significantly straight contour segments) on a contour. The success of the algorithm in identifying faces is the more creditable when it is noted that the algorithm appears to have little or no tendency to suggest spurious major faces. In ten profiles, showing eighty four 'major' faces only one has been missed and two identified only partially. In further support of the algorithm it should be noted that the missed face lay on a curved contour segment and was not a true crystal face (subhedral olivine [5.4.2.8]).

The analysis of minor faces and subfaces by the algorithm appears to have been less successful. Table 5.13 suggests false negative and false positive rates of approximately 30%. These could be viewed as inappropriately high error rates with which to characterise the algorithm when the subjective nature of the task and the difficulty in setting up a 'consensus' solution with which to compare the analysis are considered. Taking the positive view, it could be argued that the algorithm's performance is broadly comparable to the individual human opinions used to build the consensus solution.

The ambiguity associated with the identification of minor faces reduces the value of the performance evaluation. Nevertheless the testing discipline has been extremely useful in demanding a close examination of every face or subface suggested by the multiscale analysis, and of every face or subface in the consensus solution. This testing has clearly highlighted two areas of weakness in the multiscale analysis algorithm as it currently stands: the greedy selection process and the final filter pass.

For several test profiles, e.g. allanite, topaz, and subhedral olivine, failure to find faces (false negatives) has been traced back to the unwanted deletion of those faces during the greedy selection process. Much less often the deletion occurred in the final filtering pass, e.g. in the lamprophyllite example. The production of unwanted suggestions for faces (false positives) has sometimes been traced to a failure to remove digitisation 'step' faces in the final filter, e.g in the first augite example. Possible alteration to the greedy selection process and final filtering pass will be discussed in proposals for further work below [5.5.1].

The test examples above have also demonstrated that the faster arc length progression is in general inadequate for producing a useful contour analysis. When the contours show relatively little noise the faster progression may be acceptable, as in the topaz and nosean examples. For contours with larger levels of noise the faster progression is likely to be inadequate. This has been demonstrated, for example, with the allanite, the lamprophyllite, both olivines, both augites, the heptagon, and the arrow test profiles.

It may also be noted that the example analyses produced by Lowe's algorithm have shown how inappropriate it is to try to find possible crystal faces on the basis of local contour straightness measures alone.

5.4.4 Computational aspects

In addition to effectiveness, the efficiency of the algorithm must be examined. There are two aspects which may be considered: how long the algorithm takes to run, and how much space it requires. It will be seen that in considerations of both the time and space requirements of the algorithm it seems appropriate to analyse the algorithm in two separate parts: the creation of the intermediate representation, and the processing of the intermediate representation to find possible faces and refine this set of faces into the final face set. In the first stage of processing the space and time requirements of the algorithm are functions of the length of the contour chain code. The space and time requirements of the second stage are functions of the number of faces found, althought the dependence is less well defined and other factors are involved.

Considering first the generation of the intermediate representation, some rough *a priori* estimates of complexity can be made. If the contour length, N, is assumed to be a power of 2 then, in the case of the quicker (doubling) arc length progression, the actual arc lengths used will be 2, 4, 8, ..N/2. Thus the number of arc lengths used is $\log_2(N/2) = \log_2 N-1$. The size of the mmspace array required is therefore roughly $N\log_2 N$, i.e. contour length times the number of arc lengths used.

The time complexity could be expected to be a function of N.2+N.4+..+N.N/2, i.e. N.(2+4+..+N/2), as each arc-chord calculation is repeated around the contour. Since (2+4+..+N/2)=N-2, an $O(N^2)$ time complexity would be expected. In the case of the slower arc length progression the number of arc lengths used would be approximately twice that of the quicker progression, doubling storage requirements. The time complexity would be approximately doubled, but still $O(N^2)$.

The memory requirements of the second phase of processing clearly depend on the number of faces found, since virtually the only additional storage required is for the face records. The time complexity will also depend on the number of faces found, but the relationship will depend on other unpredictable and probably unquantifiable factors, and there seems little point in attempting an *a priori* estimate. The time will depend on the geometric relations between the faces, and not simply their number. For example, a set of multiply overlapping faces would generate much more work than a set of disjoint faces.

Since the behaviour of the algorithm could not be predicted running time information was recorded for a version of the program applied to each of the above test examples. The version of the algorithm which was monitored used three density slicing passes and the processing route did not treat perfectly straight faces as a special case [5.3.3.5]. The time of each run was recorded as two values: (a) the time needed to create the intermediate representation, and (b) the time needed to analyse the intermediate representation to produce the three sets of possible face records, to reduce this by the greedy selection process and to adjust the face endpoints to give a strict hierarchy. The times did not include the final pass to remove possible poor faces from the face tree [5.3.4.4].

The results presented below represent the analyses of 16 test profiles, each analysed using both the faster and slower arc length progressions.

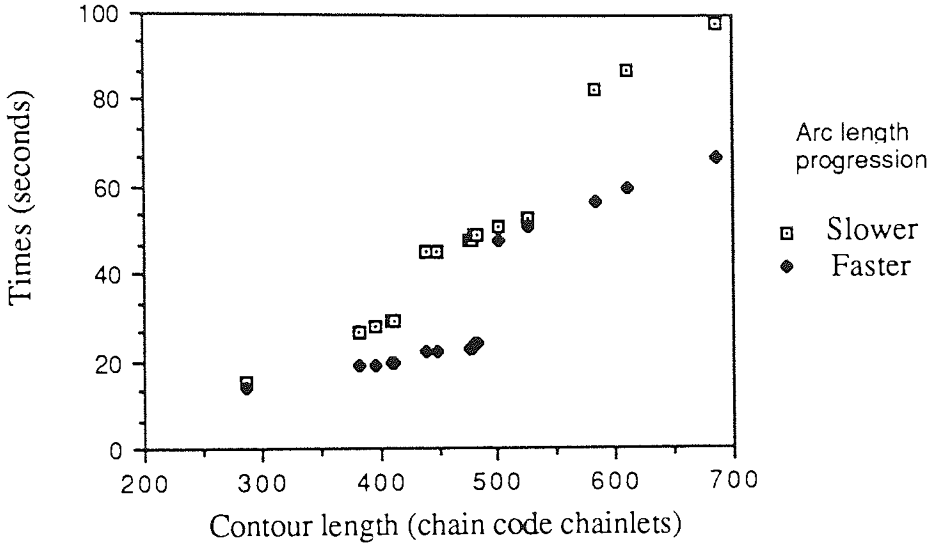


Figure 5.73 Time *v* contour length, for creation of intermediate representation

Figure 5.73 presents in the form of a scatter plot the times of the first phase of the algorithm. Data for both the slower and faster arc length progression are shown in the diagram. The data for the faster arc length progression show a sudden increase in times around the contour length 484 chainlets. Since the maximum arc length used must exceed 0.53 of the contour length [5.3.3.1], this is the point where the new arc length of 256 chainlets is used in the calculations and approximately doubles the processing time. Similar jumps in times would be expected at contour lengths 242 and 966, where arc lengths 128 and 512 are introduced, but no data was recorded around those values. The slower arc length progression shows similar steps, as each new arc length is introduced.

It is hardly justifiable to extrapolate from such a small data set, but on the basis of this evidence the time complexity for the generation of the intermediate representation appears to be a linear function of N. Thus the O(N^2) estimate given above does not appear to underestimate the complexity.

Figure 5.74 shows the times for the second phase of the algorithm, presented as a scatter plot of time against number of face records input to the greedy selection procedure. This diagram represents analyses where both faster and slower arc length progressions were used, since the method of creating the intermediate representation is of little importance at this stage. As can be seen, the time is not a unique function of the number of faces. The time is affected by how the faces overlap and how many faces are eliminated by the greedy selection, among other factors. Sixty four examples were analysed to produce figure 5.74 (i.e. sixteen profiles, each analysed with fast and slow arc length progressions, and with two processing passes: first using a special category for perfectly straight faces in the density slicing and a second time without [5.3.4.2]).

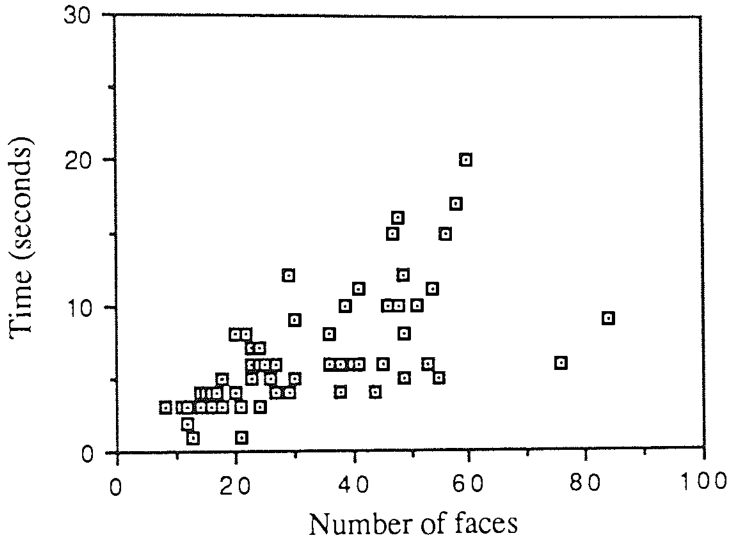


Figure 5.74 Timings for second phase of algorithm

This sample is considered too small to support any reliable extrapolation, but it does suggest that an approximate upper limit on the time complexity is given by a linear function of the number of faces. It therefore seems as though a conservative estimate of the algorithm time complexity is $O(N^2)$,where N is the contour length measured in chaincode chainlets. The algorithm's space requirements appear to be $O(N\log_2 N)$.

5.5 Conclusions and further work

The examples presented above have demonstrated the strengths and weaknesses of the new multiscale analysis algorithm. In this section the directions of further improvements and development will be discussed, and some conclusions drawn about this approach to contour analysis.

5.5.1 Further algorithm development

The evaluation of the algorithm carried out above has identified some aspects which could be improved. The greedy selection procedure currently used has been seen on occasion to lead to the loss of some minor faces; the final pass to remove poor faces has sometimes been found to be insufficiently strict [5.4.3]. These weaknesses probably account for the majority of the false negatives and false positives found in the evaluation.

The problem with the greedy selection has been seen to be that occasionally overlapping contour segments representing possible faces are found on adjacent rows of mmspace, and are therefore placed in competition with each other. The selection process then deletes the shorter face(s). One way of preventing this would require accurate measurements of the widths of the faces being placed in competition. If a face were found to be too coarse, i.e.with too high an aspect ratio, then it could be deleted at this stage. If the faces were found to be of significantly different lengths and yet feasible faces, then they could both be retained, regardless of their position on adjacent mmspace rows.

The final filtering pass also appears to need slight adjustment. The rule which expects 'step' subfaces to represent no more than 15% of their containing face was meant to err on the side of caution, but would now appear to be too biased against rejection of unwanted faces.

Another addition to the analysis which has already been suggested is a final processing pass over the face tree in order to find a better estimate of the orientation than that currently provided, which is the direction of a straight line segment joining the points which represent the extremes of the face. This additional processing would only be required for the noisier faces. The suggested solution is to find the best fitting line, in the least square sense, to the contour segment bounded by the endpoints of the face marker. Details of the calculations involved are readily available in the literature. Pavlidis [1977, Ch.7] presents the expressions for the angle and distance from the origin which the optimal line segment must satisfy; Press *et al.* [1988] supply other algorithmic details.

In addition to providing an improved estimate of the orientation of the face, which is required for subsequent symmetry analyses, this calculation would also provide an accurate estimate of the noisiness of the face, in the form of the error of the best fit line, which should then be normalised by the face length. This value could be used to quantify the quality of development of the associated crystal face.

The primary aim of the investigation described in this chapter was to find an algorithm which satisfied the design requirements, and the question of efficient implementation was of secondary importance. Future work should, however, include an investigation of how the efficiency of the algorithm can be improved, without loss of accuracy. A suitable area for examination would therefore be the most computationally expensive part of the algorithm, which is the production of the intermediate representation: the calculation of the arc-chord distances which make up the three-dimensional wedge of e(p,m,k) values described above [5.3.1]. When the l2 error norm is used to analyse a segment it is possible to build upon values produced in the analysis of subsegments, a technique used by Davis [1977a], for example. No such computational trick appears to be available when the l_∞ norm is used, as in creating the intermediate representation. Some other simplification must be sought.

One possibility is to use coarser sampling in the production of the arc-chord distances, by moving along the arc more than one point at a time, for example with a step size proportional to the arc length. No investigations have been carried out, but it seems likely that the accuracy of the algorithm could be impaired by such a modification, if the steps used did not in some way adapt to the structure of the contour. A more promising approach, therefore, is to use results from finer resolutions in the construction of the coarser resolution data.

The scheme which is proposed for further investigation would use fine to coarse resolution, i.e. an increasing sequence of arc lengths, say $k=k_0$ to k_M. Assuming that the computations for arc length k_i are about to be made, the rows of mmspace and cpspace for all arc lengths k_0 to k_{i-1} will be available. These arrays would be used to locate critical (corner) points between which the contour is relatively straight. Only these critical points would then be used to produce the k_ith plane of e(p,m,k) values, and hence the k_ith rows of mmspace and cpspace.

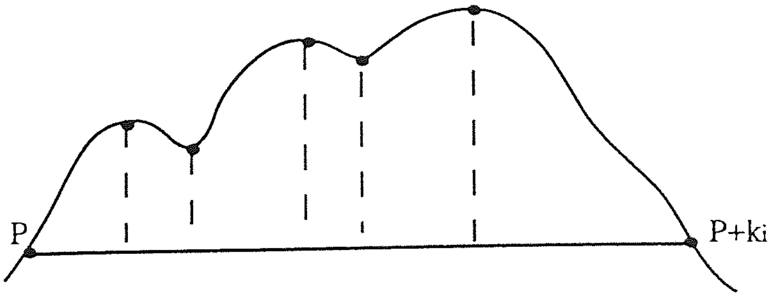


Figure 5.75 Simplifying arc-chord computations

Figure 5.75 illustrates the idea: the arc and chord from p to $p+k_i$ are being considered. Instead of calculating the arc-chord distance for every point on the arc only critical points located by previous, shorter arc lengths are used. Linear interpolation can be used for the non-critical points. Such a modification should give a significant reduction in the computation involved. This has similarities with the approach used by Bengtson [5.2.2.5]

If this approach should prove unsuccessful - and even if it were successful -

another possibility is to seek a parallel implementation. The process of calculating e(p,m,k) values appears to have a high level of inherent parallelism.

5.5.2 Summary and conclusions

In this chapter the development of a new algorithm for contour analysis has been described in detail, from a consideration of the initial motivation, through its design and implementation to evaluation and testing.

Two requirements for useful crystal profile analysis have been identified. First, since the optimal resolution cannot be predicted and may vary around the contour, the analysis must operate at a range of resolutions and avoid the use of arbitrary error thresholds. Second, some mechanism should be provided which supports the use of non-local shape information such as symmetry or collinearity. Thus, when ambiguity exists regarding the best set of faces on the contour, the additional non-local information can be used in making the final decision. This would provide an implementation of a simple example of the Prägnanz principle: the final figure has the highest pattern 'goodness' [Biederman *et al.* 1991]. It would appear that this approach has not previously been developed in practical schemes for contour analysis.

A review of contour analysis algorithms found none which provided a complete solution to the problems of crystal profile analysis. A new algorithm was therefore devised. The provenance of this algorithm has been explained: ideas have been drawn together from the work of Davis, Lowe, Phillips and Witkin, and several original contributions have also been made. The face-tree data structure has been proposed as a succinct representation of the significant linear structure on a contour in a convenient form for further analysis. A multiscale analysis algorithm which produces a face-tree representation of a contour was then presented.

The algorithm achieves its analysis via an intermediate representation of the contour which quantitatively describes the 'corners' and 'sides' on the contour over a range of resolutions. This intermediate representation is novel, and its generation and properties have been described in detail. An original procedure for analysing the intermediate representation has been presented: a density slicing technique which identifies possible faces on the contour, and a greedy selection process to refine the set of suggested faces. One novel feature of this process is its ability to cope with a variation in optimal resolution around the contour.

A procedure for eliminating the partial overlap between faces to produce a strict hierarchy of faces and proper subfaces has been presented. This is also original. Criteria for eliminating poor subfaces have been given. The implementation of all the principal procedures used in the analysis have been described in detail to allow technology transfer. Possible further improvements to the analysis scheme have been discussed.

Considerable attention has been paid to the evaluation of the algorithm. A new scheme for evaluating the effectiveness of the algorithm has been developed, extending a current method used to analyse the performance of simple classification algorithms. This evaluation has supported the view that the algorithm performs well. Major faces are detected with low error rates, and without producing spurious suggestions. Minor faces are handled slightly less well, with occasional recognition failures, and a tendency to produce too many suggestions for insignificant faces. The causes of these failings have been identified, and their correction discussed.

This chapter has therefore made theoretical and practical contributions to two-dimensional shape analysis, introducing new ideas and new algorithms. The work has been directed towards the analysis of crystal profiles. The problem of finding significant linear structure in contours is, however, not unique to petrography.

Finding such structure within contours is a fundamental step in the analysis of an image and is a key step towards understanding the structure of the objects and surfaces represented, and their spatial relationships. The algorithm may therefore be found valuable in other applications.

The consideration of symmetry analysis in the following chapter forms a natural continuation of the work of this chapter.

CHAPTER 6

Symmetry Analysis

6.1 Introduction

This chapter continues the development of the petrographic analysis scheme outlined in chapter 3 by considering the symmetry properties of crystal profiles. Some problems and possibilities of symmetry analysis are considered, and some novel algorithms developed.

The importance of symmetry in the analysis of crystal profiles has already been discussed in detail [3.5.2.3]. The symmetry of a crystal is a crucial property in classification and identification. At a more fundamental level, symmetry has been seen to be a key factor in identifying characteristic faces in polycrystalline rock sections and unravelling the effects of occlusion.

When the assignment of an interface is uncertain, say a relatively straight interface between crystals A and B for which convexity gives no information, one approach to the problem is to consider each face on the interface and estimate which symmetries associate it with crystal A and which associate it with crystal B. The face, and hence the whole interface, is assigned to the crystal which produces the more symmetrical unit. This requires a method of finding symmetries and evaluating their significance.

Another task of symmetry analysis is to complete the process of face-finding. In the previous chapter a scheme has been presented which produces a set of possible faces for a crystal profile. It is possible that more than one level of significant structure can be found, in which case the suggested faces are presented in the form of a face tree. Associated with each face is a measure of its 'width' based on the breadth of a minimum covering rectangle or perhaps produced by a least squares fitting procedure [5.5.1]. The face length and width measures can be used to derive a measure of significance. When faces contain subfaces of similar significance this reflects an uncertainty in choosing the final set of faces. In such situations a symmetry analysis can be made, directing the choice of faces towards those which form the best symmetric unit.

These symmetry analyses subsume the basic task of providing a description of the symmetry of a crystal, required for classification or identification. If symmetry analysis for face assignment is performed then the symmetries found to be associated with groups of faces may be stored, so choosing the final set of faces will allow the corresponding set of symmetries describing the profile to be accessed.

Symmetry analyses are designed to operate on the symbolic representation of the petrographic image stored in the SPIN database following the multiresolution analysis described in the previous chapter. The representation of the face data in the database will therefore be described before consideration of its use. A procedure to prune the face tree of unwanted subfaces which is appropriate for use in crystal profile analysis will also be considered.

6.1.1 Database representation of faces and polygonal approximations

Multiresolution contour analysis can produce for each crystal profile a 'conventional' fixed error polygonal approximation [5.3.5] and a list of possible faces [5.3.4.3]. These descriptions must be added to the spatial information database. New database tables are required: the polygon table, the faces table and the face tree table.

Each tuple in the polygon table stores the fixed error polygon approximation of a region contour:

```
polygon_tuple = [region_label no_vertices
                                [vertex_triplets]]

vertex_triplet =   [chainlet_no  xco  yco]
```

no_vertices is the number of vertices in the approximating polygon. The elements of the vertex list are three-element sublists which each give the position of a vertex as the corresponding chainlet in the contour's chaincode representation, and as x and y coordinates.

As has been previously described [5.3.4.3], the possible faces found by the multiresolution analysis form a face tree: if two faces overlap by more than just their endpoints then one must be a proper subface of the other. The above description of the multiresolution analysis did not include the creation of an explicit tree structure; the face records were held in a simple list. The creation of the tree structure will be described below. These faces are represented in the SPIN database using two tables: the first stores the face attributes and the second records any hierarchy among the faces.

The tuples of the faces table have the form:

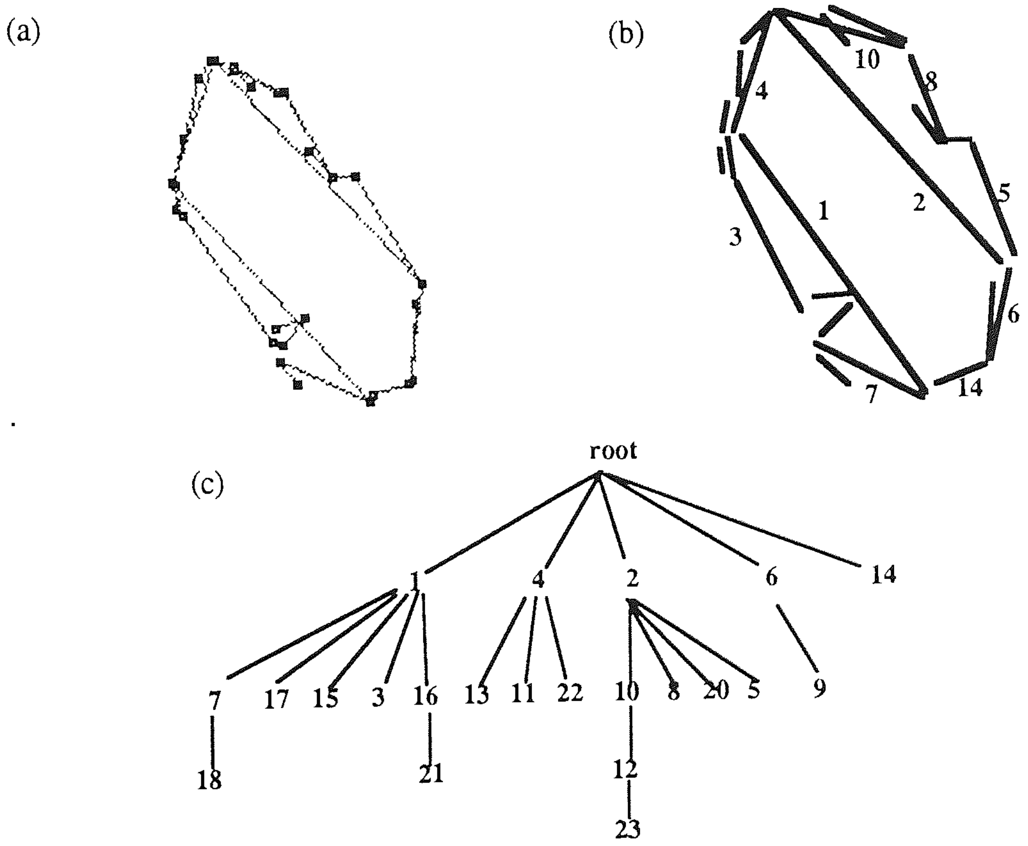
```
faces_tuple = [region_label  no_faces [face_records] ]

face_record = [ face_id    normal_id
                start_ch    startx    starty
                end_ch     endx      endy
                pseudolen   truelen   max_width
                aspect    norm_dir ]
```

no_faces gives the number of suggested faces on the profile. The face_records list of the faces_tuple has one sublist per face, recording the face attributes. face_id is a label identifying the face uniquely among others on this region profile. norm_dir is the direction of the outward facing normal associated with the face. normal_id will be described below [6.4]. The remainder of the face attributes have been described in the previous chapter [5.3.4].

The tuples of the face tree table indicate the hierarchical structure of the hypothesised faces:

```
face_tree_tuple   =   [region_label    root_face_list ]
face_sublist_element   =   [ face_id    face_sublist ]
```

The elements of the root_face_list are two-element sublists. The first element of the sublist is the face_id identifying a face; the second element is a list representing the subfaces of this face. The nested structure of the lists reflects the hierarchy of the face tree. Figure 6.1 provides an illustrative example.



(d) [1 [7 [18 []] 17 [] 15 [] 3 [] 16 [21 []]]
4 [13 [] 11 [] 22 []]
2 [10 12 [23 []]] 8 [19 []] 20 [] 5 []] 6 [9 []]   14 []]

Figure 6.1 A face-tree of faces and the database representation

Figure 6.1(a) shows the final set of possible faces for the topaz crystal profile produced by the multiresolution analysis; figure 6.1(b) is a sketch of figure 6.1(a), showing the labels given to some of the more important faces. Figure 6.1(c) makes the hierarchical structure explicit. Figure 6.1(d) is the database `root_face_list` representing this hierarchy.

The nested list representing the face tree was constructed from the face record data using a recursive POP-11 algorithm. First the face records were sorted in order of decreasing length. In order to remove problems of wrap-around it is convenient to take the chainlet corresponding to the start of the first face in this list as an origin, and adjust other chainlet values accordingly. This gives a linear ordering on the contour. The endpoint of the longest face is a suitable choice for origin since this point cannot be internal to any face in the face list.

A recursive procedure may then be applied to find subfaces of a face. Each invocation of the procedure calls itself up to three times: first to extract the longest subface, then to extract possible faces in the two gaps between the limits of the longest subface and the limits of the original face. The sublists from the calls are assembled into the compound sublist which is passed up to the calling procedure. The procedure is first called with the limits of a 'pseudoface' covering the complete contour. Once a subface has been identified in the ordered list it is removed from further consideration by the algorithm, reducing searches and preventing infinite recursion. It may be noted that there is an implicit insertion sort within this algorithm. As a result of this, the elements of each face's subfaces list are ordered as they appear on the contour, as can be seen in the example of figure 6.1.

6.1.2 Pruning the face tree

As described in the previous chapter [5.3.4.4], the multiresolution analysis algorithm contained a supplementary filtering pass which aimed to remove unwanted minor 'subfaces'. This filter was designed primarily to remove the 'step' faces on straight oblique faces, introduced by the digitisation process. Some attempt was made to extend the scope of the filter to allow it to remove other unwanted faces. However, as the examples of section 5.4.2 showed, this filter was 'cautious' in deleting faces, reflecting a wish not to risk the removal of a valid subface.

The supplementary filter presented in the previous chapter aimed to be suitable for use regardless of the application domain, and it was noted that an additional filtering pass could be made which was tailored to the specific problem domain. The problem of further pruning the face tree of unnecessary faces in a manner appropriate for crystal profile analysis will now be considered.

The pruning process devised was executed in two passes over the face tree.

The first pass removed all subfaces of faces which were narrow (width less than 2) or simply 'good' (aspect ratio less than 1/15). It also removed subfaces which were of similar width and direction to their parent face. This pass resembled a harsher version of the supplementary filter used in the multiresolution analysis [5.3.4.4].

The second filtering pass was concerned only with faces with one subface. Two conditions were checked for: subfaces which were so long as to represent a near duplication of their parental faces, and subfaces which were too poor to represent credible alternatives to their parental faces. The first of these conditions was already checked for as part of the face endpoint adjustment hase [5.3.4.3]; the difference in this pass was that an absolute length difference of 5 chaincode chainlets was set and not a relative length difference as had previously been used.

The second of these conditions reflects what appears to be an important factor in perceiving a natural set of crystal faces: how much of the contour the choice of faces accounts for. A long face will not naturally be rejected in favour of a single short subface, since that leaves a portion of the contour uninterpreted. This rule has been implemented as a test on aspect ratios: solitary subfaces with greater aspect ratios than their parent faces were deleted.

Topaz

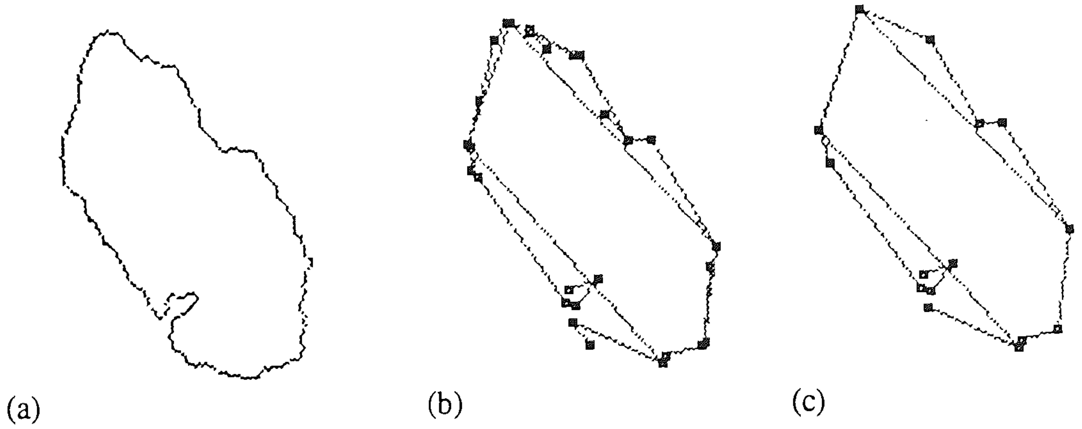


Figure 6.2 Topaz crystal profile, original face set, and pruned face set

Figure 6.2(a) repeats the topaz crystal profile of figure 5.38 and figure 6.2(b) shows the whole face-tree of faces produced for this profile by the multiresolution analysis (figure 5.39(a)). Figure 6.2(c) shows how this collection of faces has been filtered by the new pruning process, removing the duplications of the bottom right and upper left faces, and removing the subfaces which were too small to represent credible alternatives to their parent faces on the upper right and lower left of the figure. The final figure still represents a useful description of the likely faces on the contour.

Nosean

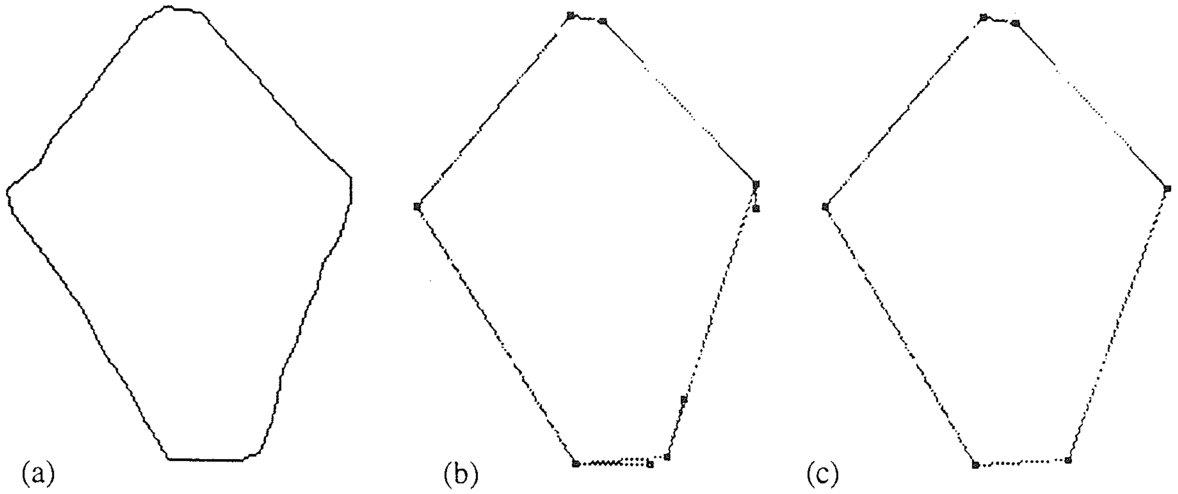


Figure 6.3 Nosean crystal profile, original face set, and pruned face set

Figure 6.3(a) repeats the nosean crystal profile of figure 5.59; figure 6.3(b) shows the final face-tree of faces produced by the multiscale analysis (figure 5.60(a)). Figure 6.3(c) shows the faces remaining after the new pruning process: the subfaces have all been removed and one simple interpretation of the contour remains.

Olivine

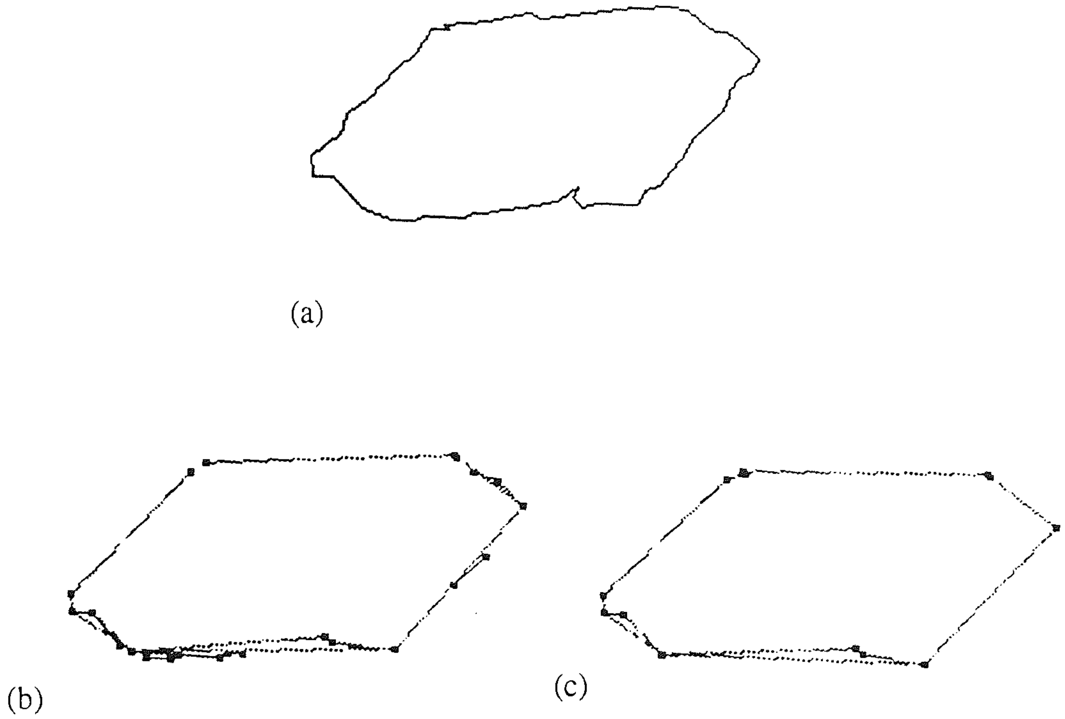


Figure 6.4 Olivine crystal profile, original face set, and pruned face set

Figure 6.4(a) repeats the olivine crystal profile of figure 5.44; figure 6.4(b) shows the final face-tree of faces produced by the multiscale analysis (figure 5.45(a)). Figure 6.4(c) shows the faces remaining after the new pruning process: most of the minor faces have been removed, giving a 'clean' description of the major faces.

Augite

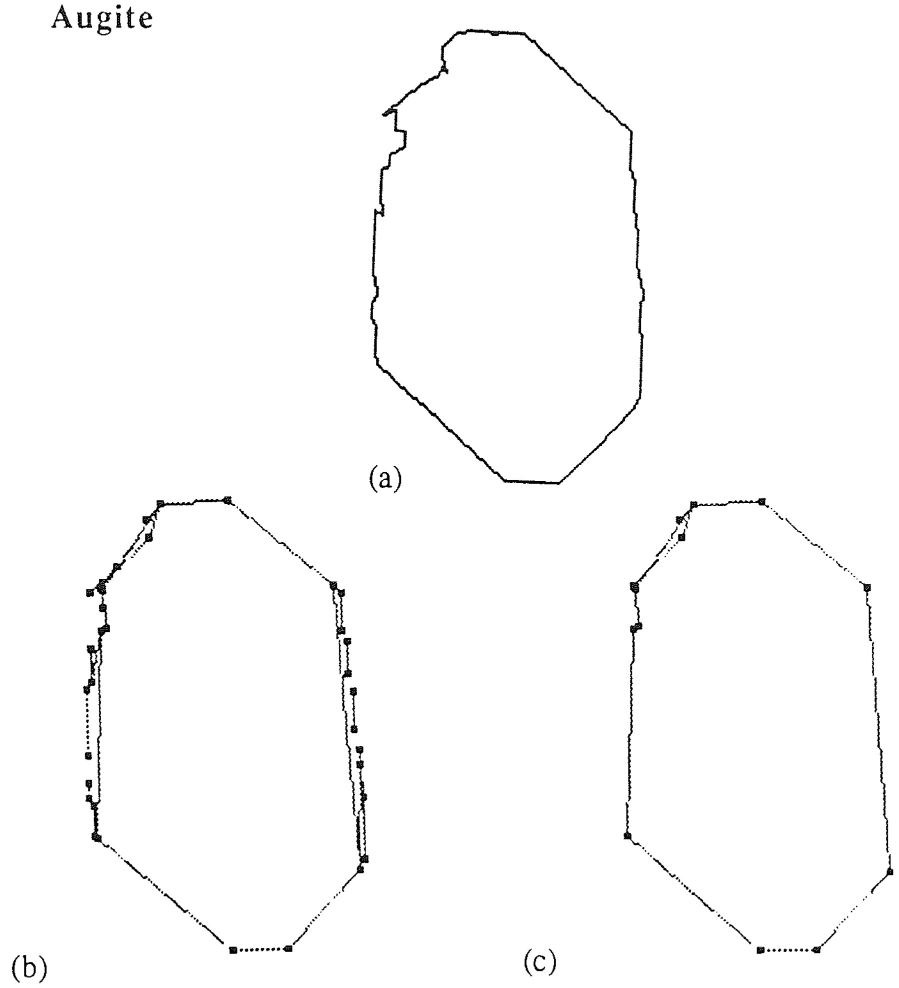


Figure 6.5 Augite crystal profile, original face set, and pruned face set

Figure 6.5(a) repeats the augite crystal profile of figure 5.50 (rotated 90°). Figure 6.5(b) shows the final face-tree of faces produced by the multiscale analysis (figure 5.51(a)). Figure 6.5(c) shows the faces remaining after the new pruning process. Again the pruning has been successful in removing unwanted minor faces.

Lamprophyllite

Figure 6.6(a) repeats the lamprophyllite crystal profile of figure 5.41; figure 6.6(b) shows the final face-tree of faces produced by the multiscale analysis (figure 5.42(a)). Figure 6.6(c) shows the faces remaining after the new pruning process. Only a couple of minor faces have been deleted in this case: the pruning process might have been expected to remove more subfaces from the upper right oblique face.

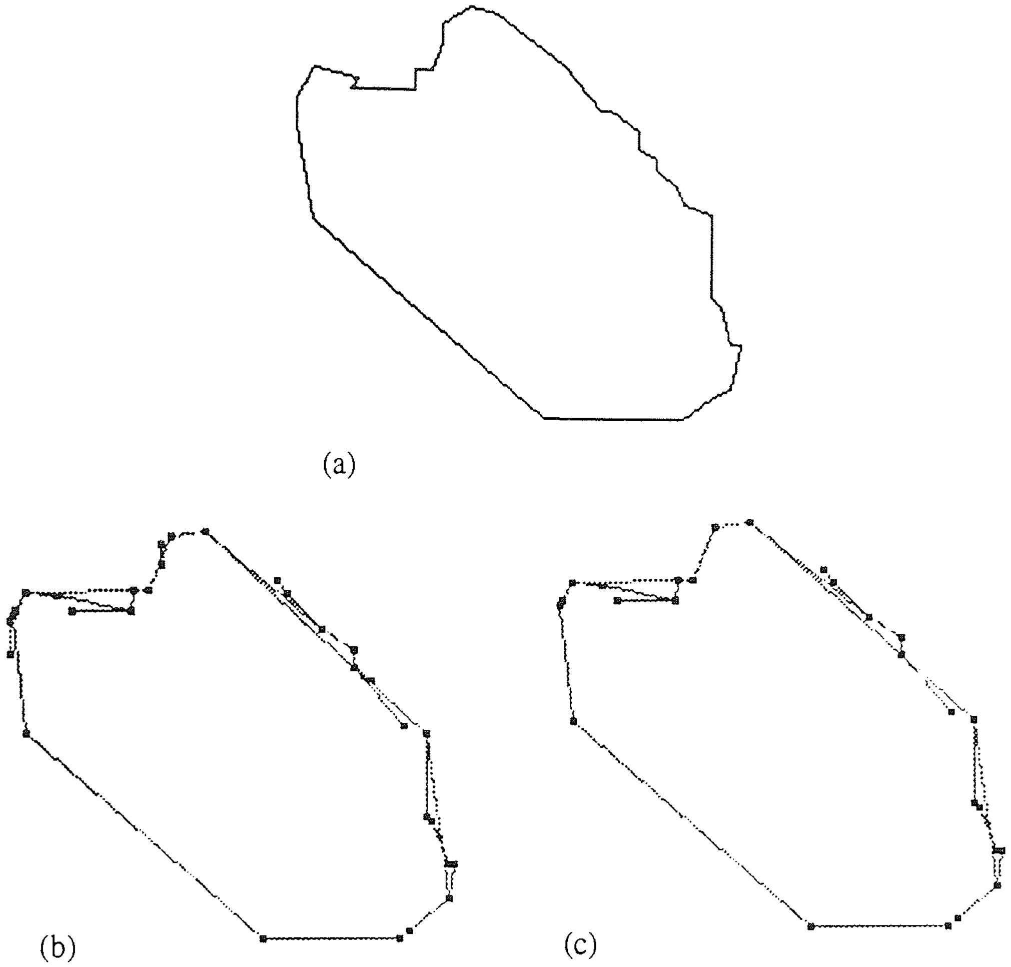


Figure 6.6 Lamprophyllite crystal profile, original face set, and pruned face set

6.2 Symmetry analysis: a review

A brief review of published work on symmetry analysis will now be presented. The description of crystal symmetry has been discussed briefly above [3.4]. There it was noted that in the study of crystals it is the internal symmetry which has paramount importance. These symmetries need not be manifested as profiles which are symmetric in the normal sense of the term. This basic difference in emphasis has meant that the published schemes for symmetry analysis which have been found do not provide suitable solutions to the problems of crystal symmetry, a fact reflected in the brevity of the review. Further review material can be found in the paper by Leou and Tsai [1987].

Davis [1977b] developed methods of detecting approximate local symmetries of a polygon, and constructed a hierarchical representation of these symmetries. Wechsler [1979] extended this approach, decomposing shapes into symmetric subparts, where each subpart could be folded along a mirroring axis. The axes were piecewise linear but not necessarily straight. The ideas of smoothed local symmetry

[Brady & Asada 1984] can be viewed as an extension of this approach to shape analysis.

Chaudhuri and Majumder [1980] found axes of mirror symmetry in two-dimensional figures using the correlation of local curvature measures. If the closed contour contained n points $\{x_i,y_i\}$, $1 \leq i \leq n$, with associated local curvatures C_i, n/2 correlations were made, corresponding to possible axes passing through the points (x_j,y_j) and $(x_{n/2+j},y_{n/2+j})$, for $1 \leq j \leq n/2$: $R_{j,j+n/2} = \Sigma_{i=2}^{n/2} C_i C_{n+2-i}$. The j values for which the correlation $R_{j,j+n/2}$ were maximal identified the positions of mirror axes. Perkins [1981] reported the use of a similar technique in a machine inspection application. Leou and Tsai [1987] used correlations to determine whether a pattern was rotationally symmetric. In such methods an axis of n-fold rotational symmetry is tested for by comparing (correlating) the contour with a version rotated through $2\pi/n$ radians.

Bolles [1979; Bolles & Cain 1983] developed a scheme for rotational and mirror symmetry analysis of two-dimensional patterns via a symbolic representation, rather than direct correlation of the patterns. Given a pattern the rotational symmetry analysis built a tree to describe it and in the process determined the rotational symmetry, which was the symmetry of the topmost feature node in the tree description. Given a tree description of a pattern the mirror symmetry analysis built a second tree to describe the mirror image and, if the two trees were similar, the patterns were declared to be symmetric. This technique used the fact that it is not necessary to locate an axis of mirror symmetry to determine whether or not a pattern is mirror symmetric.

All these techniques are concerned with 'external' rather than 'internal' symmetry and as such are not directly applicable to crystal analysis. Much more relevant is the work of Tanimoto [1980], who developed algorithms for describing the shape of polygons. Tanimoto approached curve analysis via the 'tangent indicatrix' representation, which in the case of polygons simplifies to a histogram of face directions. It is therefore similar to the 'reduced stereogram' representation of crystal profiles described above [3.4]. Clustering methods were used to group faces of the same or similar direction. Shape measures were presented for analysing these groupings. A general measure for the tendency for sides of a polygon to conform to various relative orientations was provided by the scheme of computing classes based on differences in direction measured modulo π/n, and then choosing the largest class. In particular, measures for parallelness and rectilinearity were devised, using measurements modulo π and $\pi/2$. Relevant aspects of this work will be considered again below.

6.3 The reduced stereogram

The 'reduced stereogram' representation of the orientation of features in a crystal profile has been described above [3.4]. This can be viewed as a circle upon which points mark the ends of radius vectors corresponding to the outward face normals of the crystal faces. The stereogram proper is fundamental to symmetry analysis in morphological crystallography; the reduced stereogram will be used as the basic representation of orientation data for symmetry analysis of crystal profiles.

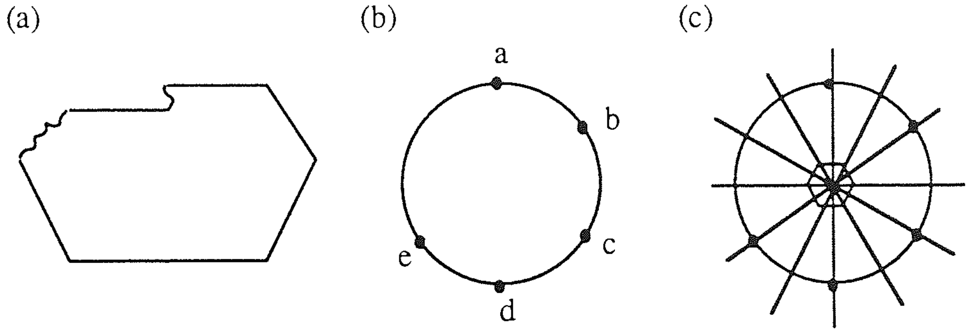


Figure 6.7 A crystal profile, its stereogram, and analysis

Figure 6.7(a) shows a crystal profile, 6.7(b) and 6.7(c) show its associated stereogram and symmetry analysis. This reduced form of stereogram records the directions of the outward face normals. Two crystal faces are irregular and have not provided reliable orientations. Only five directions are represented on the stereogram. The full circular arc is used: parallel faces on opposite sides of the crystal are represented as diametrically opposed points on the stereogram circle; collinear or parallel faces on the same side of the crystal are represented by one single point. Parallel faces with the same face normal direction will be termed ‘codirectional’; they need not be collinear. The order of faces on the contour is irrelevant; only the face directions are considered.

The analysis shows many symmetry elements: the lines represent mirror symmetries and the central symbol represents an axis of 6-fold rotational symmetry. Such an axis implies 2-fold and 3-fold symmetry also; conversely, the presence of 2-fold and 3-fold symmetry implies 6-fold symmetry. However, only 5 face normal directions have been found and so full 6-fold rotational symmetry is impossible. The analysis therefore presents some probable or possible symmetries inferred on the basis of angular differences. Here interfacial angles of 60° suggest 6-fold rotational symmetry, and it is accepted that full sets of faces cannot always be expected for real data. Six mirror planes have been hypothesised, although again not every plane successfully pairs off every face. For example, the mirror plane containing the face normals a and d does not find a face normal which is a mirror image to b.

As has been described [3.4], crystals have only a restricted set of possible internal symmetries: mirror planes, certain axes of rotational symmetry, and a centre of symmetry. As a consequence, crystal profiles can only show mirror symmetries or 2, 3, 4 or 6-fold rotational symmetry. Since the crystal section need not be perpendicular to the axis of rotational symmetry there may be a distortion of the apparent interfacial angles. Thus looking for face normals approximately 90° apart as an indication of 4-fold rotational symmetry need not succeed: for example, if a square prism is cut across a diagonal at 45° to the axis a diamond shaped section is produced with corner angles of approximately 70° and 110°. However, parallelism of faces is preserved in oblique sections, as are mirror symmetries among face normals, since these are parallelisms of interfacial angles.

The measurement of face direction on real crystal profiles is also likely to be imprecise. Assuming that the likely faces have been found by the multiresolution analysis described in chapter 6, a measure of the accuracy may be derived from the face length and the width of the rectangular covering strip. As previously described, [5.5.1] a least squares analysis may be used in some situations to give an improved orientation measure and associated error estimate.

In view of possible error and distortion it was considered that a search for 3, 4 and 6-fold rotational symmetries by seeking faces respectively 120, 90 and 60° apart was likely to be impractical. The alternative approach suggested is based on the more robust properties of parallelism and collinearity. For each crystal profile the analysis is made via an intermediate representation equivalent to the reduced stereogram.

6.3.1 Intimations of symmetry

Parallel faces on opposite sides of a crystal section are represented by a pair of diametrically opposed points on the circle of the corresponding reduced stereogram. A mirror plane is represented by a diameter of the circle; the two points at which the diameter meets the circumference give its direction, which need only be specified modulo 180°. A point on the stereogram at a mirror plane endpoint would represent a face perpendicular to the mirror plane; a point on the stereogram 90° from an endpoint would represent a face parallel to the mirror plane.

Given a reduced stereogram represention of a crystal profile, the task of finding parallel faces translates into one of checking the face normals on any semicircle of the stereogram for diametrically opposed points. Search for a mirror symmetry requires a more involved set of trials. In contrast to an exhaustive correlation approach, such as that of Chaudhuri and Majumder cited above, only a selection of mirror positions need be tested. It is only necessary to consider as candidate mirror plane orientations directions which bisect the angle between two face normals on the stereogram. For each hypothesised position further evidence can then be sought, and if none is found the hypothesis can be rejected, since for any pair of face normals a mirror plane could be found which reflects one on to the other. Supporting evidence could take the form of additional pairs of face normals which are mirrored in this plane, a face parallel to the plane, or a face perpendicular to the plane.

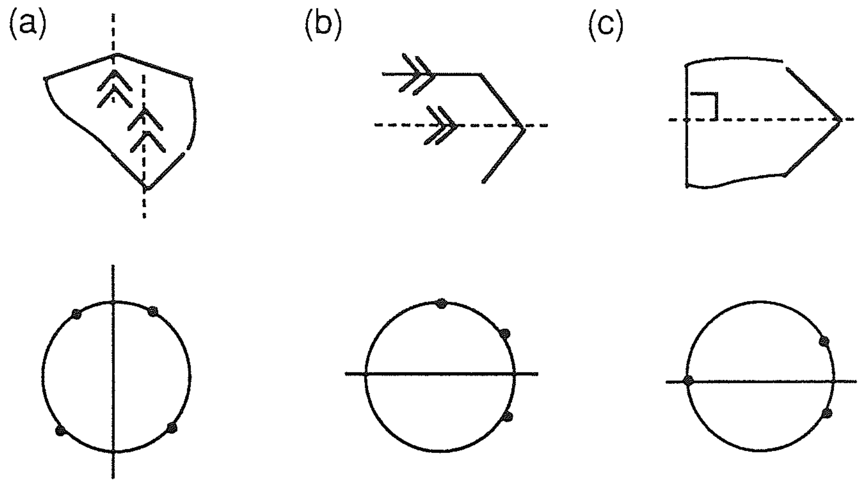


Figure 6.8 Supporting evidence for a mirror plane

Figure 6.8 illustrates these possibilities as crystal face configurations and sketch stereograms. Likely mirror planes bisecting interfacial angles have been indicated by broken lines on the crystal sketches. In figure 6.8(a) there is only one mirror plane direction, manifested as two parallel bisectors of interfacial angles. It may be remarked that the configurations of figure 6.8(b) and (c) could, for example, be seen in crystals with 6-fold and 3-fold rotational symmetry. As has been noted, the perpendicular relation between face and mirror plane is liable to angular distortion, but it has been retained since it may allow the recognition of mirror planes in some crystals.

6.4 Representation of stereogram and symmetries

Effective analysis of the stereogram requires a suitable representation. One possibility is a histogram-like array recording the occurrence of faces against orientation, probably weighted by the faces' significance measures. However, the proposed representation is a list structure which may be stored in the SPIN database, and which provides a link between face and symmetry information. Symmetries are represented explicitly in the database like other region properties.

The stereogram is represented as a list, with each element in the list representing a face or cluster of faces having similar face normal orientations. These list elements therefore represent the points marked on the stereogram circle. The stereograms will be stored in a new table in the SPIN database: the stgm table. Each tuple represents the stereogram of one profile:

```
stgm_tuple   =   [ region_label    [stgm_list] ]
```

The stgm_list is a list of sublists, each representing a cluster of faces as described above. These will be termed 'principal directions' or PDs. Each entry in the stgm_list initially takes the form

```
pd_list   =   [ pd_id   dirn   signif [member_faces] ]
```

pd_id is a unique label assigned to the principal direction, whose value is represented by dirn. The member_faces list records the labels of the faces which have been clustered to form this principal direction. signif is a measure of the significance of this principal direction, derived from the measures of significance of the faces assigned to this cluster. This is discussed below [6.4.1]. Further fields can be added to pd_list during analysis, as will also be described.

Each face record also carries a note of the pd_list of which it is a member. This is the normal_id field in the database face_record described above [6.1.1].

Symmetries

Symmetries of the profile are stored in a database table: the symmetry table. The sublists of this table each record the symmetry of a particular region:

```
symmetry_tuple = [region_label [parallelisms] [m_planes]]
```

The two sublists of symmetry_tuple record parallelisms and mirror reflections amongst the face normal directions, via the principal directions represented in the stereogram tuple. The stereogram tuple carries information on codirectional faces, since by definition codirectional faces are assigned to the same principal direction.

Parallelisms

The parallelisms list of the symmetry tuple records pairs of principal direction on the stereogram which are approximately diametrically opposed. Each parallel pairing is given a unique label, and a significance value which is a function of the significances of the principal directions involved (see below). Thus an element of the parallelisms list has the form

```
parallelism = [ par_id   pd_id1   pd_id2   par_signif ]
```

Reference must be made to the member_faces list of the principal directions involved in order to identify the actual parallel faces. Each principal direction record in the stereogram will also be extended to carry a note of any parallelism of which it is a member.

Mirror planes

The m_planes list of the symmetry tuple records the (possible) mirror planes, again expressed in terms of principal directions. Each mirror plane is given a unique label: mp_id. The sublists of the m_planes list each describe one mirror plane, taking the form:

```
  m_plane = [ mp_id  mp_dirn  [supp_pairs]  [pds_parallel]
                       [pds_perp] total_signif ]
```

mp_id is a unique label assigned to the mirror plane. The mirror plane direction is represented by mp_dirn, and should be specified modulo 180°. The three sublists record the supporting evidence for this mirror plane. The elements of the supp_pairs list are themselves three-element lists; the first two elements are the labels of pairs of principal directions mirrored in this plane, the third element is a measure of the significance of this evidence. Similarly the elements of the pds_parallel and pds_perp lists are two-element lists. The first element is the label of a principal direction respectively parallel or perpendicular to the mirror plane; the second element is a measure of the significance of this evidence (see below). total_signif represents the total of the significance values for the three categories of evidence for the mirror plane.

For easier manipulation of symmetry information a mechanism is needed to facilitate access to the symmetry hypotheses to which each principal direction contributes. This is achieved by adding four fields to the principal direction records. These then take the form:

```
pd_list = [ pd_id  dirn  pd_signif  [member_faces]
                [parallelisms] [mps] [mps_parallel] [mps_perp]]
```

These four additional lists hold the identifiers of the relevant symmetry elements. This design allows direct access to the sort of information required for appraising how a face fits into the symmetry pattern of a crystal. It also identifies the records which must be updated when a hypothesised face is removed from the database or assigned to another region.

6.4.1 Significance measures

In order to compare the importance of different symmetry features some numerical measures are required. There are many problems in deriving good measures of symmetry since several factors are involved. In this first approach to the problem the principle was followed that the importance attached to a symmetry should reflect how many of the faces were involved, and their significances. Thus parallel faces which are long and straight should provide better evidence for parallelism or a mirror plane than two short irregular faces, and be scored correspondingly higher.

The significance measure assigned to a single face was true length divided by aspect or, equivalently, the square of the length divided by the width, where 'aspect' and 'width' are as previously defined. This significance measure gives a higher value to more significant faces, in contrast to the aspect ratio measure used in the previous chapter. In order that the value does not become unbounded a minimum width value of 1 is used. It may be noted that the absolute length of the face is used, and since this is measured in pixel units the value is certainly greater than 1. If the face length were expressed as a fraction of the perimeter then squaring would no longer be appropriate.

To derive an appropriate measure for a principal direction representing a set of faces the lengths of the member faces were summed and the widths were averaged, using a weighting corresponding to the face lengths. Thus if two faces of lengths l1 and l2 and widths w1 and w2 were assigned to the same principal direction the weighted average width would be w=(l1.w1+l2.w2)/(l1+l2). The final significance measure is then (total length)^2/(weighted average width).

The significance measure assigned to a parallelism between two principal directions was the minimum of the two significance values involved. The minimum was used to prevent small noisy faces parallel to good faces appearing too significant. By the same reasoning the significance assigned to a mirrored pair of principal directions was the minimum of the two values concerned. If more than one pair of principal directions were reflected the values from each pair were summed to give the overall significance value.

If a principal direction was found perpendicular or parallel to a mirror plane the overall significance value was increased by adding the minimum of the significance of this principal direction and the maximum significance value amongst the individual pairs of principal directions. Thus, for example, if two pairs of principal directions with significances 100 and 200, and 250 and 400 were mirrored in the same plane the significance assigned to the mirror plane would be 350 (i.e. min.(100, 200) plus min.(250, 400)). If a principal direction with significance 50 were found parallel to the mirror plane orientation and another with significance 300 were found perpendicular to the plane, this would add min.(50, 200) and min.(300, 200) to the overall significance of the symmetry. The final significance value would therefore be 550.

A principal direction perpendicular to the mirror plane (representing a face parallel to the mirror plane) may already be represented in one of the pairs of reflected directions associated with the plane. This case was not handled differently: the overall significance value was incremented as for other principal directions. The double counting was used to emphasise the importance of such a configuration.

No account was taken of the degree of accuracy of the symmetries found. If principal directions differed by 180°, within a given error threshold, then they were considered parallel; otherwise they were not. The error threshold used for these experiments was 10°. This value was also used for clustering the possible mirror planes (hmps); in grouping faces to form principal directions and measuring if a principal direction were perpendicular or parallel to a mirror plane a value of 7° was used.

These error limits are unsatisfactorily *ad hoc*, intended only for early trials of the algorithms. Ideally, the fact that two principal directions were exactly 180° apart rather than at the limit of the accepted tolerance should somehow be reflected in a higher significance, probably by introducing a weighting factor which reduced significance as a function of the accuracy found. This has not been implemented, and remains a matter for future development.

In short, the significance values used were expedient measures appropriate for early experimentations only. The development of sounder measures is an area for future work. This will require more theoretical investigations; satisfactory implementation may also require more accurate face attribute measurements.

6.5 Building a stereogram

Here suitable procedures will be described for creating a database stereogram list corresponding to a set of face records. The basic problem is clustering faces with similar face normal orientations, i.e. finding the principal directions. When a non-zero tolerance is allowed in grouping faces there may be problems caused by the non-transitive nature of the relation: if faces a and b have face normal directions differing by no more than a given tolerance, and faces b and c are similarly related, this does not guarantee that the relation holds for faces a and c.

Tanimoto [1980] considered this problem in his work on polygon shape analysis. He presented two algorithms for finding clusters or 'classes' of faces, overcoming or limiting the non-transitivity problem. Tanimoto used his algorithms to construct an 'unordered class histogram'; this task was virtually identical to building a reduced stereogram representation.

In the first algorithm the classes are formed one at a time and faces are assigned as quickly as possible. The first unclassified direction is made the centre of a new class. The algorithm will be described in terms of the data structures developed above. The algorithm input is a face list; the output is a stereogram list. Initially the output list is empty. The face records in the input list are considered in turn. The face normal direction is compared against the principal direction associated with every class in the output list. If the directions coincide to within a given tolerance T the face is assigned to this class by having its `face_id` label added to the `member_faces` list. If the face normal direction is not within the tolerance of any class yet established then a new class is created, with a new label, with this face's direction as the principal direction, and this face's label is added to the `member_faces` list.

This algorithm is described as "quick and dirty" since in each case the first unclassified face normal direction is taken as the central, principal direction of a new class, which may lead to 'unnatural' classes.

Tanimoto suggested a solution to this problem in his second algorithm, which used a more sophisticated clustering technique. The algorithm is 'furthest neighbour' clustering [Duda & Hart 1973, sect.6.10], viewed as a problem in graph construction. Each face direction is made a node in a graph, and arcs connect pairs of faces whose orientation difference is within a given tolerance. The arcs are also labelled with this angular difference. Clusters of faces where non-transitivity of the grouping is present are represented in the graph by components which are not complete subgraphs, i.e. there exists at least one pair of nodes in the subgraph not connected by an arc. The algorithm therefore checks each component for completeness, and splits incomplete subgraphs by deleting the arc representing the largest angular difference. Several deletions may be required to produce a set of complete subgraphs. The resultant subgraphs are the required classes. The principal direction of the class may be taken as the average direction of the faces in the class.

Clearly other clustering techniques may be applicable. This is an area for future investigation. Time constraints did not allow the implementation of these more sophisticated algorithms. However, the 'quick and dirty' classifier was implemented, with a modification which improved its performance. Instead of allowing the faces to be presented to the classifier in a random order the input sequence was ordered by the associated significance values. This improved the chance of establishing reasonable choices of orientations for the cluster central values, at relatively little computational cost.

Illustrative example: building a stereogram

This algorithm was applied to a set of faces taken from a multiresolution analysis of the crystal profile shown in figure 6.17(a). Figure 6.17(b) shows the faces found by the analysis, following the pruning described above [6.1.2]. Figure 6.17(c) indicates the subset of these faces selected for the stereogram construction, and their labels. Figure 6.17(d) shows a reduced stereogram representation of these faces.

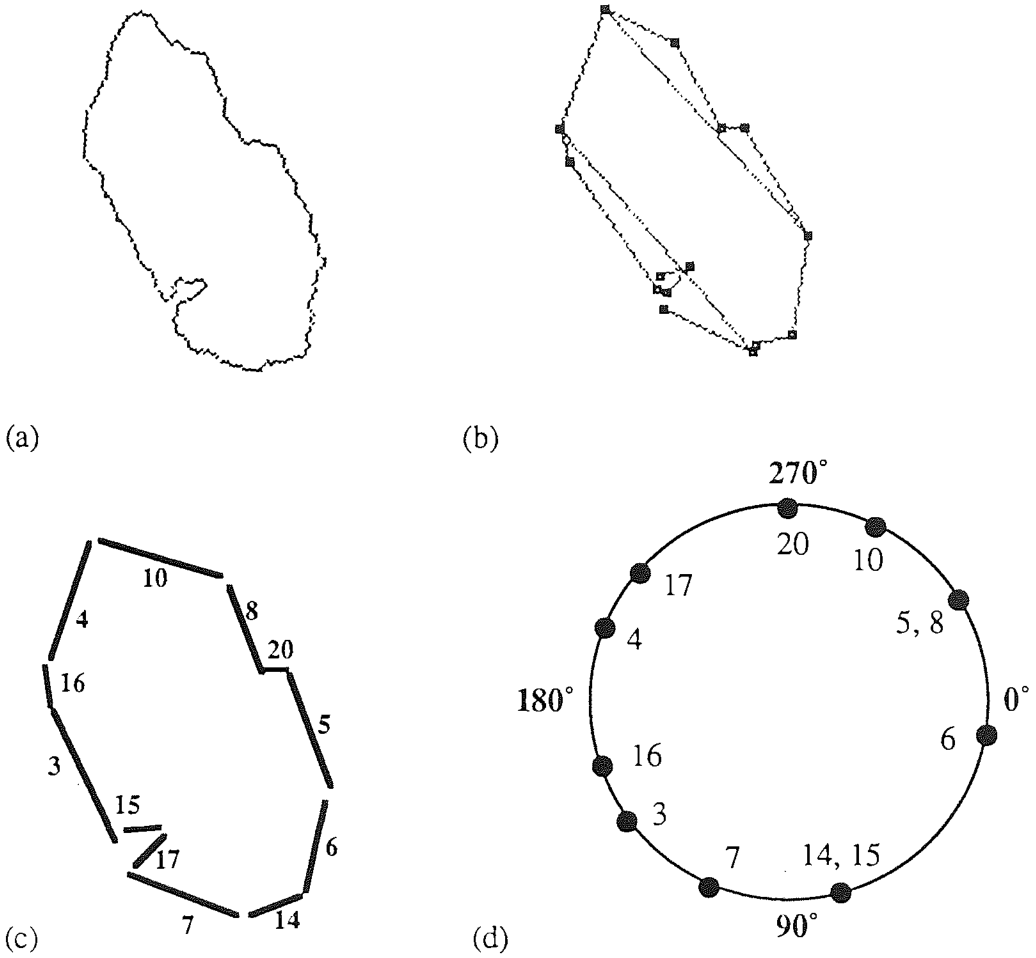


Figure 6.9 A set of crystal faces and associated stereogram

From figure 6.9(d) it can be seen that ten principal directions have been found for the twelve faces. These are marked as points on the circle; the labels of the faces represented are given at each principal direction.

P.D.	angle	significance	faces assigned
22	163	54	16
20	221	55	17
23	75	253	15, 14
16	295	128	10
14	116	194	7
12	9	281	6
5	146	660	3
10	330	1418	8, 5
9	201	389	4
25	270	49	20

Table 6.1 The stereogram list for the face set of figure 6.17

Table 6.1 lists details of the stereogram principal directions. The entries are: principal direction label, principal direction value in degrees (figure 6.9(d) shows the clockwise angular convention used), the significance value, and the labels of the faces assigned to the principal direction.

6.5.1 Conflicting interpretations

The stereogram of figure 6.9 represents only a subset of the faces in the tree for this profile. This is a maximal self-consistent subset: no face is a subface of another and, given this restriction, no part of the contour has been left unrepresented unnecessarily.

It is possible to produce a stereogram representing all the faces in the face tree, but this must be done in an appropriate manner. The purpose of the stereogram is to represent the clusters of face normal directions in a form suitable for further analysis. Therefore care should be taken to avoid inappropriate classifications such as a face A being assigned to a cluster whose principal direction has been defined by a face B whose existence is inconsistent with that of A, e.g. A is a subface of B.

One possible approach to avoiding these problems is to generate every possible maximally consistent set of faces from the face tree and to treat each set separately, producing a stereogram and, subsequently, symmetry analyses.

6.5.1.1 Generating maximal consistent sets of faces

Figure 6.10(a) sketches all the possible faces found by the multiresolution analysis of the crystal profile of figure 6.9(a), with some minor faces removed by the pruning process described above [6.1.2]. Figure 6.10(b) shows the corresponding face tree.

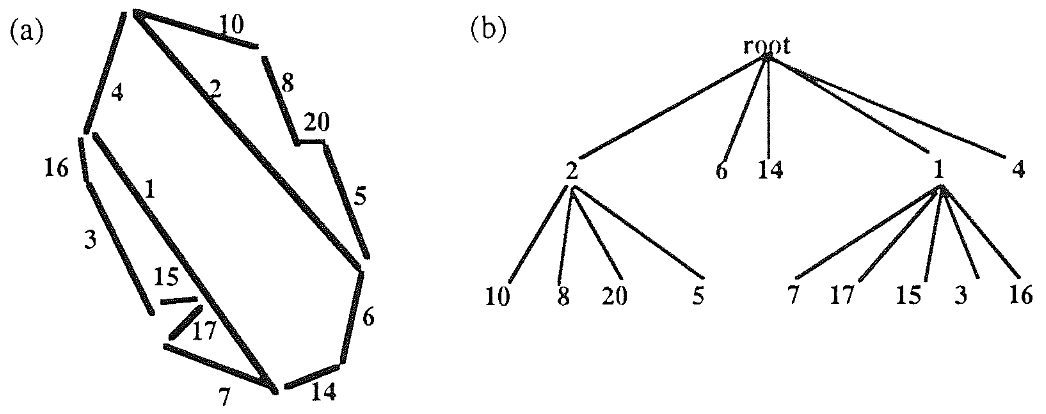


Figure 6.10 Pruned face tree for topaz crystal profile

If a face is selected as a valid face then none of its subfaces can be chosen, since each would give a rival interpretation for some part of the contour. Similarly, no face containing this face can be selected. However, all other subfaces of the containing face (i.e. the siblings in the face tree) which have no subfaces must be selected, since they do not conflict, and to omit them would leave a portion of the contour unnecessarily uninterpreted. Thus, with the face tree of figure 6.10, if face 10 is selected then so must faces 8, 20 and 5; if face 3 is selected then again so must faces 7, 17, 15 and 16. If the sibling faces have subfaces it would also be acceptable to replace one or more of them with their subfaces.

All maximal consistent sets of faces may be generated from the face tree using a recursive series of rewriting processes. Each tree node label, initially holding only the identity of the face represented, is replaced by a list of sequences of labels representing all the possible consistent maximal subsets of faces in the subtree below this node. Figure 6.11 illustrates the operation of this algorithm.

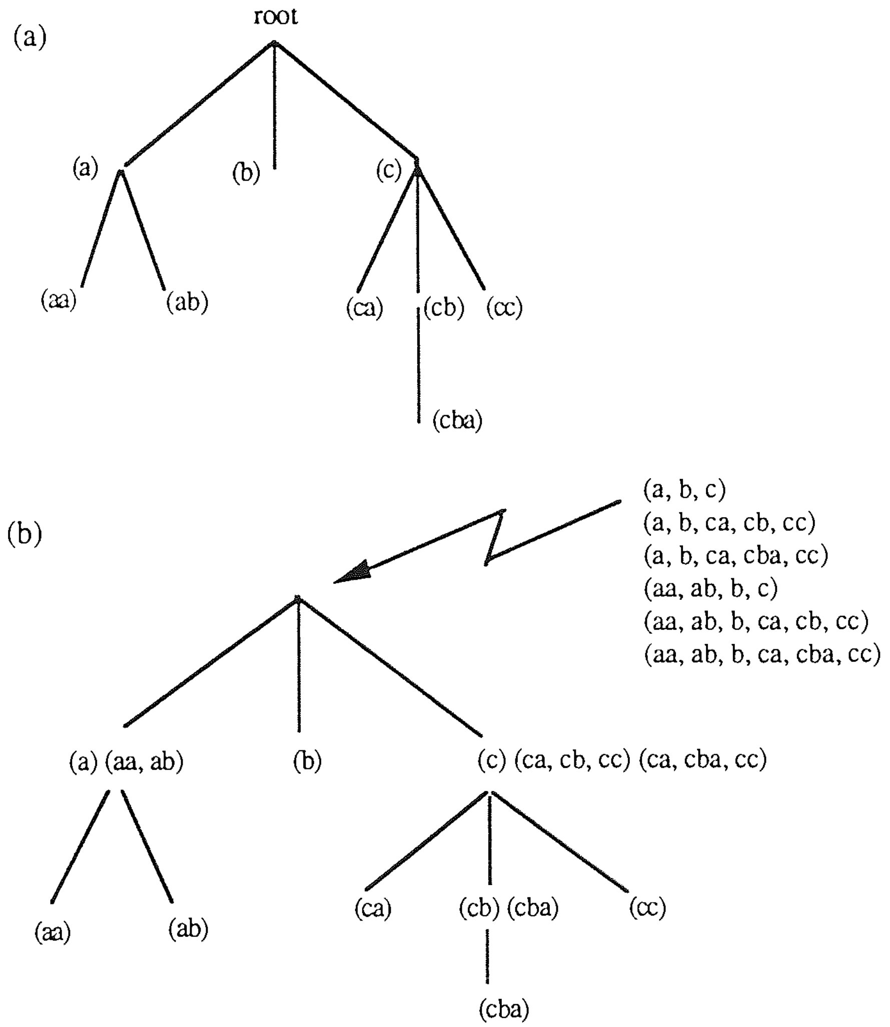


Figure 6.11 Generating maximal consistent face sets

Figure 6.11(a) represents a face tree with face labels at the nodes. Figure 6.11(b) shows the lists of labels associated with the nodes as the recursive tree traversal unwinds. The label of a leaf node is unchanged; the label of a parent node with one child becomes augmented by the list passed up by the child; the label of a node with two or more children becomes augmented with all valid combinations of the lists passed up by the children. In this example the recursion terminates when the root contains a list of six node sequences, representing the six possible maximal consistent face sets.

This algorithm, like the others discussed in this chapter, has been implemented in POP-11. The maximal consistent face sets produced by applying the algorithm to the face tree of figure 6.10 are:

(a) [7 17 15 3 16 4 10 8 20 5 6 14]
(b) [7 17 15 3 16 4 2 6 14]
(c) [1 4 10 8 20 5 6 14]
(d) [1 4 2 6 14]

The stereogram construction described above used the maximal face set (a). It may be noted that the unpruned version of this face tree (figure 6.1(b)) would generate 80 maximal face sets - showing the desirability of removing unnecessary faces at an early stage of processing.

6.5.1.2 Stereogram construction using the face tree

Generating and analysing every maximal consistent face set could involve unnecessary duplication of work. For example, suppose ten faces were found on a profile and one had a subface: this would lead to the generation of two maximal sets, each of ten faces, but with nine faces in common. There ought to be a way of avoiding the redundant effort in processing the same nine faces twice.

A stereogram could be formed using the nine common faces and then each of the two conflicting faces could be classified and either assigned to an existing principal direction or made the centre of a new one. However, this approach is flawed in view of the clustering mechanism used: one of the two faces may be most significant and find itself assigned to a principal direction whose orientation value is not as accurate as it could be, since it has been defined by a less significant face. On the other hand, if the face were made the centre of the cluster it may later be rejected in favour of its subface, and so the principal direction's central value would no longer represent any valid face on the contour.

These problems can be overcome if, when constructing the stereogram list, careful use is made of information about the face tree structure. For example, the faces can be presented to the classifier in order of significance, as before. When a face f is presented to the classifier the face tree is examined to see whether f lies in a subtree of any face already classified, or contains any face already classified. If so, f cannot be clustered with that face since the two faces cannot coexist in any consistent face set for the contour. If necessary, therefore, face f must be made the centre of a new cluster to avoid an inappropriate pairing. If face f is to be assigned to a cluster whose direction has been determined by a face g which the tree reveals to be liable to replacement by a subface or containing face then, as well as assigning face f to the cluster based on g, a new cluster should be formed around f. If f itself is liable to replacement then a version of any cluster f is assigned to should also be retained without f as a member.

These steps may result in two or more classes representing the same or similar orientations; any further faces presented to the classifier with a similar direction must be assigned to all the appropriate classes. In practice relatively few new clusters are likely to be generated.

The above is an informal description of a suitable strategy for building a stereogram. The algorithm has been implemented in a different manner, however, since there is no need to repeatedly analyse the tree to discover the allowable combinations of faces: all the information required can be found in more accessible form in the list of maximal consistent face sets. The clustering algorithm implemented will now be described. The stereogram produced is termed the 'master stereogram' for the profile.

6.5.1.3 Building the master stereogram

The input data required are the face list and list of maximal consistent face sets. The faces are considered in order of significance. Each face direction is compared with all the principle directions represented in the stereogram, which is initially an empty list. If the directions do not match, to within a given tolerance, for any of the principal directions in the stereogram, then a principal direction record is created for the face and added to the stereogram.

If the face direction is sufficiently close to a principal direction represented in the stereogram then further tests must be made. Using f to denote the face to be assigned, and M to denote the `member_faces` list of the principal direction to which f could be assigned, the maximal face sets are searched for instances of three key situations involving f and M.

The first significant sort of maximal consistent face set sought is one which contains every element of M but which does not contain f. If such a maximal face set is found it means that a version of this principal direction must be retained in the updated stereogram without f being assigned to it, since the maximal face sets have shown that face f must be susceptible to replacement by a descendent subface or ancestor face above it in the face tree. Whether a version of the principal direction should be created with f assigned to it depends on the presence of the other two key situations.

The second significant sort of maximal face set sought is one which contains f yet contains no element of M. The existence of such a maximal face set appears to imply that a new principal direction must be created for the stereogram, based on f. However, it is only after all the principal directions in the stereogram have been examined, and it has been found that no element of any of the member_faces lists is ever found in a maximal face set together with f, that a new principal direction record should be created for f, and the stereogram updated.

The third significant sort of maximal face set sought is one which contains f and every element of M. If such a face set is found it means that f is compatible with all the other faces representing this principal direction. The stereogram should therefore be updated with a version of the principal direction record with f assigned to the member_faces list.

The algorithm has been implemented and examples of its operation are given below. For the tree of faces shown in figure 6.10 the resulting stereogram is given in table 6.2. One cluster has been formed for faces 15 and 14, centred on face 14, and another cluster consists of face 14 alone, since face 15 is liable to replacement by face 1. Faces 5 and 8 have been clustered together, with the cluster centred on face 5, the more significant face. Neither face 5 nor 8 is assigned to a separate cluster, since face 5 is replaced by face 2 if and only if face 8 is.

P.D	angle	significance	faces assigned
25	270	49	20
9	201	389	4
10	330	1418	8, 5
5	146	660	3
2	319	845	2
4	140	505	1
12	9	281	6
14	116	194	7
16	295	128	10
23	75	253	15, 14
18	75	99	14
20	221	55	17
22	163	54	16

Table 6.2 Master stereogram representing figure 6.18

As a result of this modified classification each principal direction record represents a consistent set of faces. The importance of this modified clustering will become clearer when the stereogram is used for symmetry analysis [6.6.3].

6.6 Implementation of symmetry analyses

Symmetry analyses are made via the stereogram representation described above, although the basic procedures could be modified to analyse face records individually. The analyses implemented have already been outlined: the detection of parallelisms and mirror symmetries [6.3.1]. As has been observed, these analyses can also detect some significant configurations produced by rotational symmetries.

First the analysis of an unambiguous crystal profile will be described, i.e. a set of faces with no conflicting levels of structure. The analysis of a master stereogram representing conflicting sets of faces will then be considered.

6.6.1 Analysis of simple face sets

6.6.1.1 Parallelisms

Given a stereogram listing a set of principal direction records, finding parallel faces translates into a search for principal directions whose angular difference is 180°, within some tolerance T. This search is implemented by forming two groups of principal directions and examining all possible pairs with members drawn one from each group. Since a non-zero tolerance is used these two groups must overlap slightly.

First the stereogram records are ordered by their principal direction values. The first group represents directions in the range [90-T, 270+T]; the second group represents directions in the range [270-T, 90+T], accounting for wraparound. For each record in one set every member of the other set is examined to see if the principal directions have the appropriate difference. For each parallel pair found a 'parallelism' record is created for the symmetry tuple [6.4], storing the identities of the two principal directions and the significance of the parallelism [6.4.1].

6.6.1.2 Mirror planes

The first step in finding mirror symmetries is to form a list of hypothesised mirror planes (hmps) by taking each unordered pair of principal directions in the stereogram and forming a four-element hmp record. The elements of this record are the angle of the mirror plane which would reflect the principal directions on to each other, the labels of the two principal directions concerned, and a measure of the significance. The angle of the mirror plane is stored modulo 180°. In practice only principal directions at least 40° apart in orientation were considered: it was considered that a genuine mirror plane was unlikely to be found between two faces with a small interfacial angle, and so this opportunity was taken to reduce the number of hmps generated. The significance values were used to order the hmp records in preparation for the second phase of the analysis: the accumulation of evidence.

The hmps must be clustered so that hmps with similar orientations are brought together. This was done using the same 'quick and dirty' classification process as for the stereogram construction. The hmps were presented for classification in order of significance to improve the chance of establishing reasonable centre of cluster orientation values. It is of course possible to cluster without using the significance values, so their presence at this stage of the algorithm can be considered optional. In the clustering process the basic `m_plane` records described above were established [6.4], with the pairs of principal direction labels associated with each hmp being stored in the `supp_pairs` sublist.

When every hmp has been assigned to some class, further evidence is sought in the form of parallel and perpendicular principal directions. For every cluster of hmps, the stereogram is searched for principal directions whose orientations are parallel or perpendicular to the hypothesised mirror plane, to within a given tolerance. The labels of the principal directions found are stored in the appropriate `pds_par` and `pds_perp` lists of the `m_plane` tuple.

When all supporting evidence has been gathered hypothesised mirror planes with insufficient supporting evidence are deleted: a mirror plane is removed if its sole support is one pair of faces. It may also be possible to set a threshold on significance at this point to remove weak symmetries, but such a scheme was not implemented. The surviving mirror planes are then assigned unique labels.

6.6.2 Illustrative examples

6.6.2.1 Topaz crystal analysis

Table 6.3 presents the results of the analysis for mirror planes in the set of faces shown in figure 6.9(c). The corresponding stereogram is shown in figure 6.9(d) and listed in table 6.1 The analysis are given in terms of principal direction labels, not face labels; reference must be made to the stereogram listing or figure 6.9 to identify the faces concerned. Figure 6.12 presents some of these results in the form of reduced stereograms.

Format of mirror plane records [6.4]:

```
m_plane     =   [mp_id mp_dirn [supp_pairs] [pds_parallel]
                     [pds_perp] total_signif]

supp_pair   =   [pd_id1  pd_id2  signif]

pd_perp     =   [pd_id  signif]

pd_parallel   =   [pd_id  signif]
```

[11 58° [[10 5 660.1] [12 14 193.6] [10 22 54.2] [16 22 54.2] [25 9 49.0]
[25 20 49.0]] [] [[10 660.1] [5 660.1]] 2380.3]

[10 86° [[9 10 389.4] [5 12 281.3] [14 23 193.6] [10 20 54.6]
[16 20 54.6] [12 22 54.2]] [[25 49.0]] [] 1076.7]

[9 174° [[9 5 389.4] [5 20 54.6] [14 20 54.6] [25 23 49.0]] []
[[25 49.0] [23 253.2]] 849.8]

[8 105° [[9 12 281.3] [5 23 253.2] [12 20 54.6]] []
[[9 281.3] [12 281.3]] 1151.8]

[7 138° [[9 23 253.2] [23 20 54.6] [14 22 54.2] [25 12 49.0]]
[[5 253.2]] [[20 54.6]] 718.7]

[6 22° [[10 23 253.2] [14 16 128.4] [25 5 49.0] [25 14 49.0]]
[[9 253.2]] [[14 193.6] [16 128.4]] 1054.8]

[5 42° [[12 23 253.2] [10 14 193.6] [5 16 128.4] [25 22 49.0]]
[[20 54.6]] [] 678.8]

[4 158° [[9 14 193.6] [12 16 128.4]] [[10 193.6] [22 54.2]] [[23 193.6]] 763.4]

[3 68° [[9 16 128.4]] [[23 128.4]] [[10 128.4] [22 54.2]] 439.4]]

[2 5° [[16 23 128.4] [20 22 54.2]] [[12 128.4]] [25 49.0] 360.0]

[1 119° [[23 22 54.2] [25 10 49.0]] [[14 54.2] [16 54.2]] [[9 54.2]] 265.8]

Table 6.3 Results of mirror plane analysis on figure 6.9

Figure 6.12(a) shows the principal directions of the stereogram and their labels. Figures 6.12(b) and (c) show the two most significant mirror planes found: these mirror planes are numbered 11 and 8 respectively in table 6.3. The planes are marked as diameters of the stereogram circle and the mirrored pairs of face normal directions have been linked by broken lines. Double headed arrows mark the principal directions which have been found perpendicular to the mirror plane; the principal directions which have been found to be approximately parallel to the mirror plane have not been marked in any special way. Figure 6.12(d) repeats the original contour.

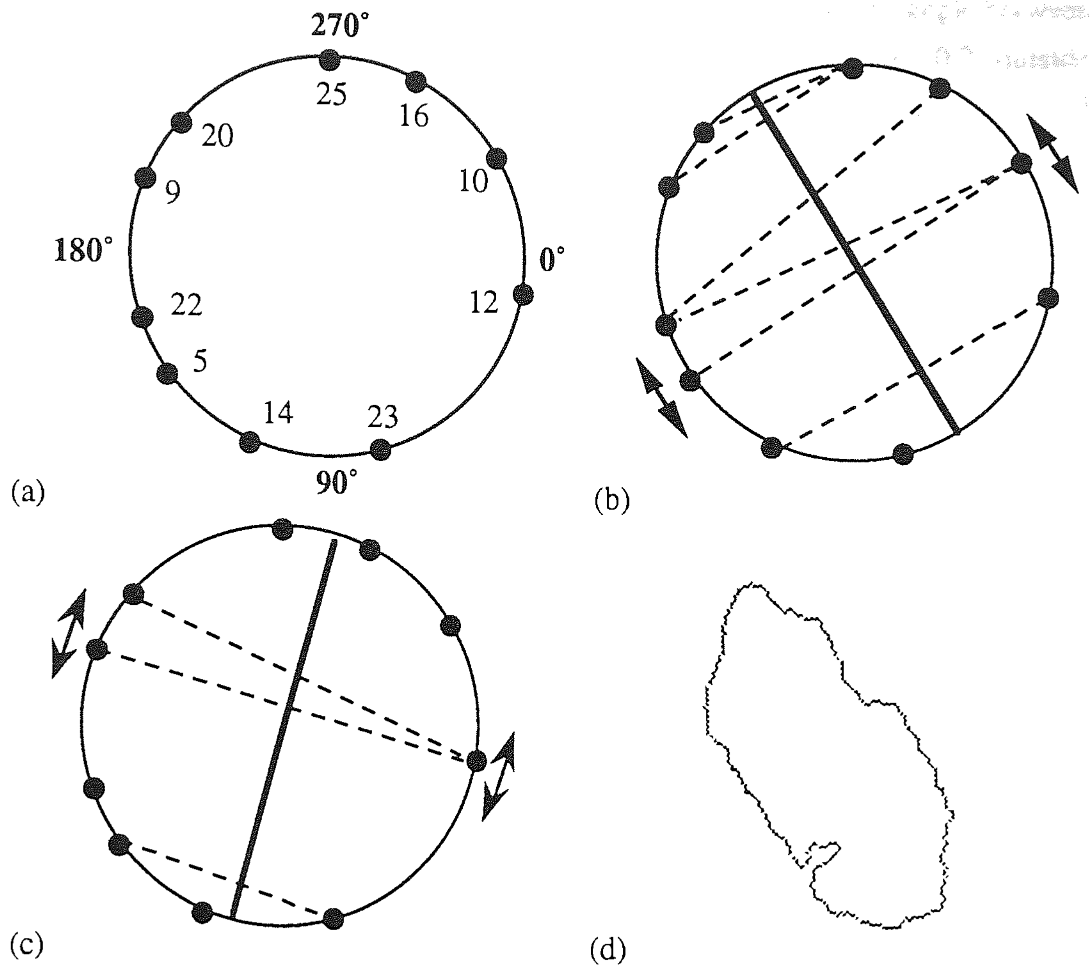


Figure 6.12 Stereogram and the two major mirror planes found for topaz crystal

It may be noted that the mirror pairings may sometimes appear inconsistent: in figure 6.12(b), for example, principal direction 22 is paired with both directions 10 and 16, yet these two principal directions differ by 35°. The hmp angle for pair 10 and 22 is 66°; the hmp angle for pair 16 and 22 is 49°. Thus they are both within the 10° tolerance of the mirror plane's central orientation of 58°, defined by the pair of principal directions 5 and 10. Such pairings could perhaps be removed but it was felt that they did help to indicate possible symmetry elements and they were retained.

Only two parallelisms were found: between faces 7 and 10 and between face 3 and the codirectional pair 5 and 8, i.e. between principal directions 14 and 16, and between principal directions 5 and 10.. The angular difference between faces 4 and 6 (principal directions 9 and 12) was ~168°, more than the tolerance limit of 10° from 180°.

As an examination of the crystal profile shows (figure 6.9(a)), the most evident mirror plane has been found by this analysis to be the most significant symmetry element. Table 6.3 records that the mirror plane oriented at 58° bisects the angle between the normals of faces 7 and 6, 3 and 5, 3 and 8, 16 and 5, 16 and 8, 16 and 10, 4 and 20, 4 and 17, 14 and 12. In addition it is parallel to faces 3, 5 and 8. It may be noted, however, that the mirror plane just fails to bisect the angle between the normals of faces 4 and 10: the angular difference is 10.2°, i.e. only 0.2° outside the tolerance limit set. This suggests that the simple tolerance threshold used in these analyses should be modified, changing the sudden cut-off into a more gradual decline; this matter will be considered below [6.7.1].

6.6.2.2 Nosean crystal analysis

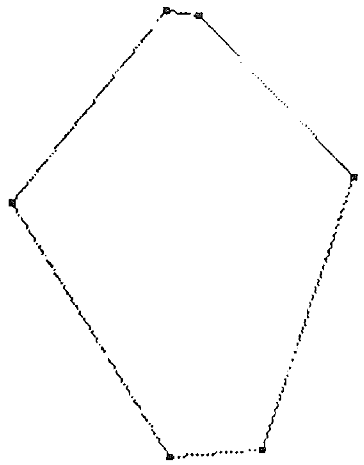


Figure 6.13 Faces found on nosean crystal profile

It was shown above [6.1.2] that pruning unwanted subfaces from the face tree produced by the multiresolution analysis of the nosean crystal profile resulted in one consistent set of faces. Figure 6.13 reproduces figure 6.3(c), showing the six remaining faces. Table 6.4 lists the details of the associated reduced stereogram. The face labels assigned, in clockwise order from 12 o'clock, were as follows: 5, 2, 3, 8, 4, 1.

P.D.	angle	significance	faces assigned
12	279	163	8
10	85	550	5
8	218	2576	3
6	316	2948	4
4	18	2513	1
2	148	3887	2

Table 6.4 The stereogram list for the face set of figure 6.13

Table 6.5 presents the results of the analysis for mirror planes applied to the nosean stereogram data. Figure 6.14 presents some of these results in the form of sketch stereograms.

Format of mirror plane records [6.4]:

```
m_plane = [mp_id  mp_dirn  [supp_pairs] [pds_parallel] [pds_perp]  total_signif]
supp_pair =     [pd_id1  pd_id2  signif]
pd_perp   =     [pd_id  signif]
pd_parallel =   [pd_id  signif]
```

```
[8  52°  [[6 2 2948.0] [10 4 550.0]] [] [[6 2948.0] [2 2948.0]]  9393.9]
[7  87°  [[8 6 2576.0]  [4 2 2512.6] ]  [[10 550.0]]  []  5638.7]
[6  3°  [[8 2 2576.0] [12 10 162.5]] [] [[12 162.5] [10 550.0]] 3451.1]
[5  118°  [[8 4 2512.6] [10 2 550.0]]  [] []  3062.7]
[4 167°  [[6 4 2512.6]] []  [[10 550.0]]  3062.7]
[3 152°  [[10 8 550.0] [12 4 162.5]] [[2 550.0]]  []  1262.6]
[2 21°   [[10 6 550.0]] [[4 550.0]]  []  1100.1]
[1 33°  [[12 2 162.6]] [[8 162.6]  []  325.1]
```

Table 6.5 Results of mirror plane analysis on figure 6.13

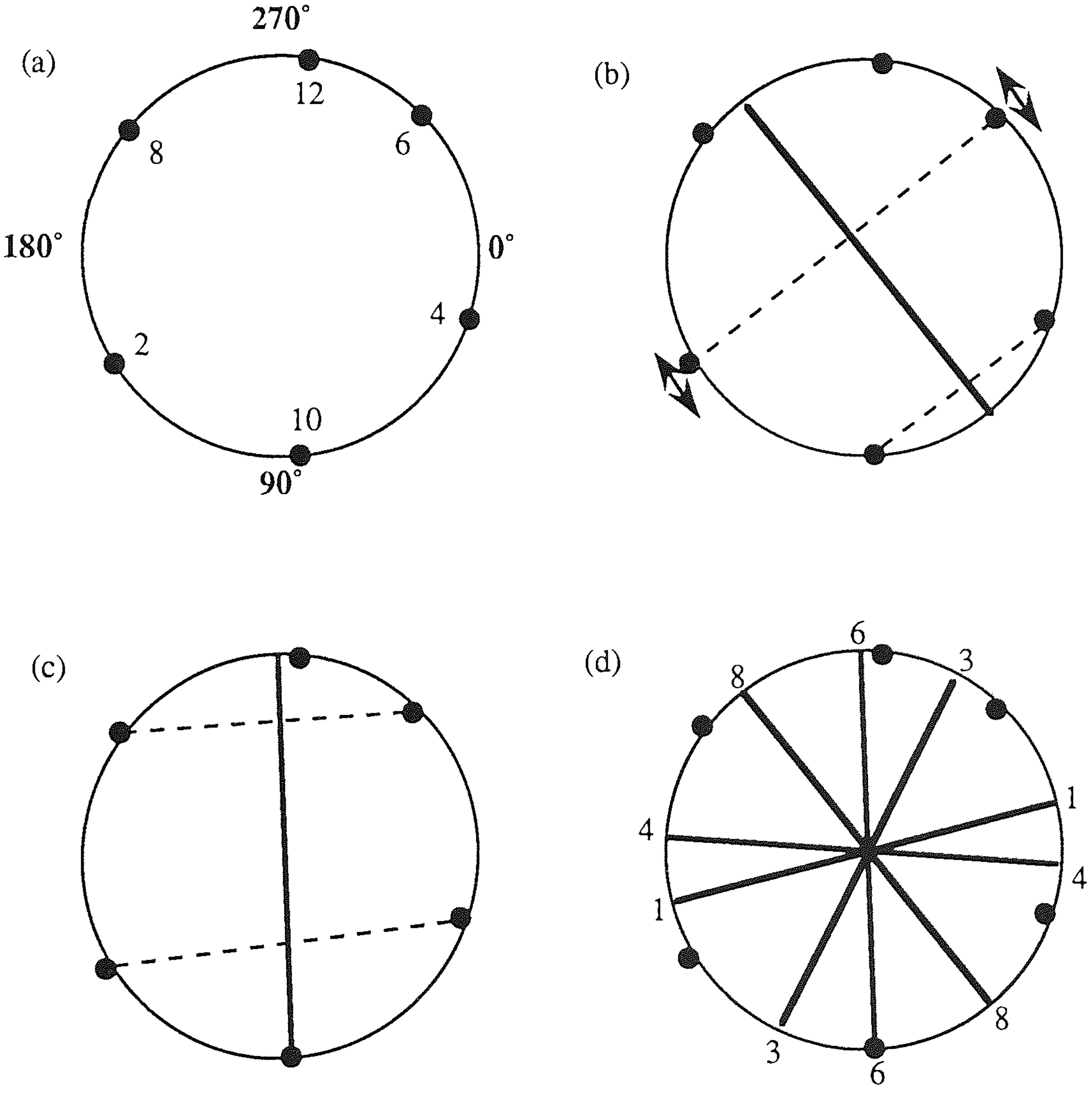


Figure 6.14 Symmetry analysis of nosean crystal

Figure 6.14(a) shows the reduced stereogram for the nosean profile, with the principal direction labels marked. Figures 6.14(b) and (c) show the two most significant mirror planes found in the analysis, displayed as in figure 6.12. Figure 6.14(d) shows the directions of the five most significant mirror planes found, marked with the mirror plane labels used in table 6.5.

Probably the most readily perceived mirror symmetry in the nosean crystal profile is a vertical mirror plane, i.e at an angle of approximately 90° in the convention used here. This mirror plane has been found as plane number 7 in the table: the mirror plane at 87° has been found to bisect the angle between the normals of faces 1 and 2, and 3 and 4, i.e. stereogram principal directions 4 and 2, and 8 and 6. In addition, this mirror plane has been found approximately parallel to the normal to face 5, i.e. principal direction 10. This mirror plane is indicated on the sketch stereogram of figure 6.14(c).

The symmetry analysis has, however, suggested another mirror plane of higher significance: mirror plane number 8 in the table. This mirror plane is sketched in the stereogram of figure 6.14(b). This mirror plane, oriented at 52°, reflects the principal direction 6 on to principal direction 2, and 4 on to 10. It is approximately 16° from bisecting the angle between principal directions 8 and 12. This mirror plane is given a high significance in the analysis since both principal directions 6 and 2 are perpendicular to it, i.e. faces 2 and 4 are approximately parallel to the mirror plane. The angular separation of these face normals is 168° and so they are not found to be parallel to each other. This problem of non-transitive inconsistency has been noted above [6.5].

It is probable that the mirror plane at 87° appears to be more significant to the human observer than the mirror plane at 52° because of the better correspondences in length between the pairs of faces matched, i.e. the figure is approximately bilaterally symmetric and the 87° mirror plane is therefore very close to being a 'conventional' mirror plane. In contrast, the figure is not conventionally bilaterally symmetric about a mirror plane at 52°. This crystal profile therefore presents an example where an 'external' symmetry is readily perceived, distracting attention from an important 'internal' symmetry.

6.6.3 Analysis of face trees

In this section problems of analysing several maximal consistent face sets, all derived from the same face tree, will be considered. The likely aim of such analysis would be selecting the 'best' single set of faces present in the face tree, thus completing the face-finding process started in the multiscale contour analysis. The analysis of some real data examples will be considered, examining the relevance and effectiveness of certain strategies.

It will be seen that in certain situations it is not necessary to consider every maximal consistent face set represented by the face tree. In situations where it is appropriate to analyse several face sets derived from the same face tree there are two approaches which may be taken. The direct approach is to generate a stereogram for each face set and analyse it as described above. If the number of face sets is large, however, a more efficient approach is to build one 'master' stereogram representing the whole face tree, perform one 'overall' symmetry analysis of this stereogram, and then extract from this analysis the required information concerning each individual face set of interest.

An algorithm for the creation of a suitable 'master' stereogram has been described [6.5.1]. In this stereogram the principal directions represent only consistent groups of faces, and the the most significant face in the cluster always provides the principal direction value. These requirements may result in two or more principal directions of the stereogram representing similar or even identical face normal orientations, and some faces may be classified as a member of more than one principal direction. However, these characteristics present no real problems.

Symmetry analysis of the master stereogram can be performed in much the same manner as for 'normal' stereograms. The results relevant to any given face set can then be extracted from the overall analysis using the ideas of face-sets, face-set complements and principal direction complement sets.

A face-set is a consistent set of faces, i.e. a set of faces which do not overlap except possibly at endpoints, and no face is a subface of any other face in the set. An algorithm for generating all the maximal face-sets from a face tree has been given above; the following discussion will only apply to maximal face-sets. A face-set can be represented by a simple list of face labels, since there is no hierarchical structure to be represented.

Given a face-set the face-set complement is the set of all faces which are not members of the face-set. The principal direction complement set, or 'PD-complement set', is the set of principal directions in the master stereogram whose face lists contains at least one member of the face-set complement. The PD-complement set must also be extended to include those principal directions whose member-faces lists are proper subsets of member-faces lists of other principal directions.

The example of the master stereogram listed in table 6.2 may be considered as an example. Figure 6.10 shows the faces. If the face set selected is {7 17 15 3 16 4 10 8 20 5 6 14} then the face-set complement is {1 2}. The PD-complement set is {2 4 18}. Principal directions 2 and 4 represent faces 2 and 1 respectively. Principal direction 23 has faces 14 and 15 assigned to it and principal direction 18 has face 14 alone assigned to it. The principal direction 18 is therefore redundant for this particular face set and so is assigned to the PD-complement set.

6.6.3.1 Parallelisms

Parallelisms can be found by an analysis of the master stereogram as described above [6.6.1]. This will produce a list of all possible parallelisms. Given a face-set, the set of all parallelisms can easily be analysed to find those which concern only members of the face-set. This is most easily achieved by rejecting those parallelisms which involve a member of the PD-complement for the face-set. In this way a subset of parallelisms appropriate to each face-set can be extracted from the set of all parallelisms.

6.6.3.2 Mirror planes

The set of all possible hmps (hypothesised mirror planes) can be produced from the master stereogram as in the analysis of the simple stereogram described above. Given a face-set, the set of all possible hmps can then be filtered, rejecting any hmp which involves a member of the appropriate PD-complement set, and producing a subset of relevant hmps. These hmps can then be clustered as before, and supporting evidence sought in the form of perpendicular and parallel principal directions for each mirror plane orientation. In this search only principal directions of the master stereogram which are not in the PD-complement need be used. This will produce a set of mirror planes for the given face-set.

It may be noted that the strategy is *not* to generate a set of all mirror planes from the master stereogram and then simple filter out elements which featured members of the PD-complement. The two stage processing is required to ensure that the clusters formed in the accumulation of evidence phase are sure to have as their central mirror plane orientation a value which has been defined by elements of the set of interest, and not a value from a hmp where one of the principal directions lies in the PD-complement.

6.6.4 Illustrative examples

The scheme described above for symmetry analysis via a master stereogram has been implemented and applied to a number of crystal profiles. The examples considered below will show how the master stereogram and maximal face set lists can be used in selecting a 'best' set of faces on the grounds of symmetry. The limitations of the simple approach of comparing the symmetry 'scores' of rival face sets will be shown, and other strategies for selecting face sets will be discussed.

6.6.4.1 Topaz crystal analysis

Figure 6.15(a) reproduces the pruned tree of faces found for the topaz crystal profile; figure 6.15(b) shows the labels assigned to faces.

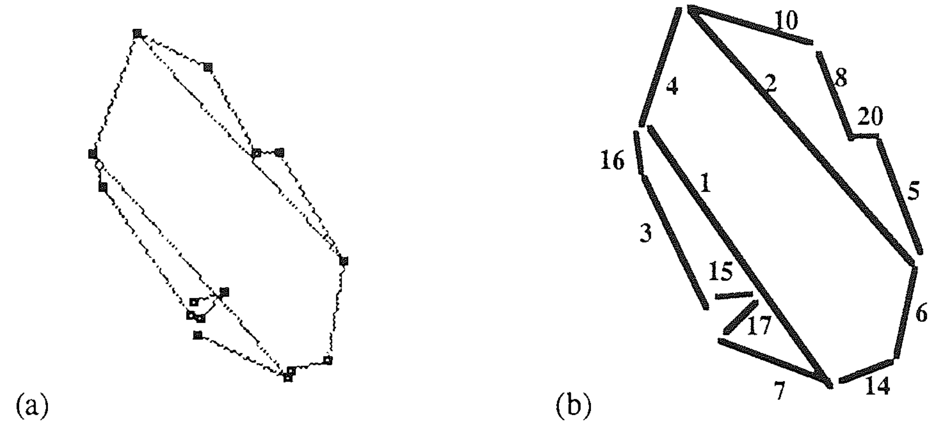


Figure 6.15 Faces found on topaz crystal profile, and their labels

a) Face set: {7 17 15 3 16 4 10 8 20 5 6 14}

Parallelisms:

faces	significance
3: 5, 8	660
7: 10	128

No. of mirror planes: 11

Total signif. measure: 9739

3 best mirror planes:

direction	significance
58	2380
105	1152
86	1077

b) Face set: {7 17 15 3 16 4 2 6 14}

Parallelisms:

faces	significance
2: 3	660

No. of mirror planes: 11

Total signif. measure: 9369

3 best mirror planes:

direction	significance
52	2035
80	1087
174	1033

c) Face set: {1 4 10 8 20 5 6 14}

Parallelisms:

None

No. of mirror planes: 11

Total signif. measure: 5509

3 best mirror planes:

direction	significance
55	1564
105	943
170	586

d) Face set: {1 4 2 6 14}

Parallelisms:

faces	significance
1: 2	505

No. of mirror planes: 6

Total signif. measure: 4688

3 best mirror planes:

direction	significance
49	1614
105	943
170	770

Table 6.6 Symmetry analyses for 4 interpretations of figure 6.15

Table 6.6 presents a summary of the results of the symmetry analysis. For each maximal consistent face set derived from the face-tree table 6.6 lists the parallelisms found, with associated significances. The parallelisms have been given in terms of the faces, and not the principal directions. The number of mirror planes found is indicated, and a summary of the three most significant planes only. A total for the significances of the mirror planes is also given.

It is most probable that the human observer would choose face set (a) as the best maximal consistent subset. The results of table 6.6 show that this particular face set yields the highest 'total significance' score for mirror planes and produces the mirror plane with the single highest significance value. This set also produces the best collection of parallelisms. Figure 6.16(a) shows the faces of set (a); figure 6.16(b) reproduces the original contour for comparison.

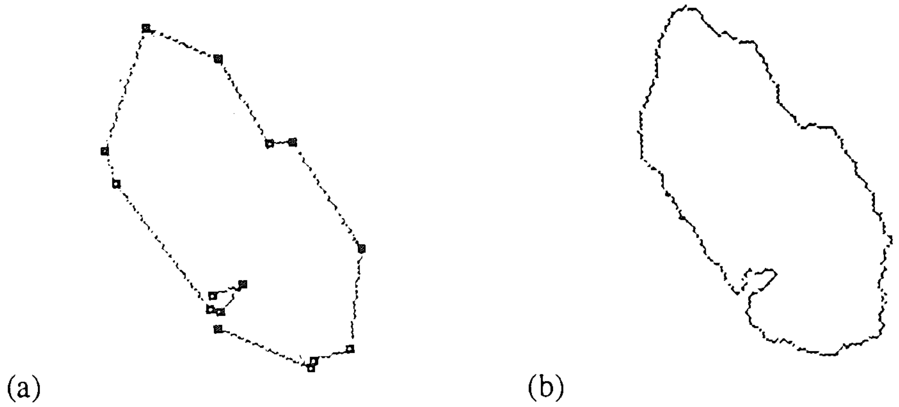


Figure 6.16 Final selection of faces for topaz crystal

The highest 'total significance score' on its own has nevertheless been found to be an unreliable indicator of the 'best' set of faces. It would appear that introducing into a face set a face which adds a new direction to the stereogram is likely to increase the total significance value through the contributions of new possible mirror planes which result. This tendency means that face sets with many members can produce high total significance scores, representing many mirror planes which are each relatively insignificant. This occurred in analysing an alternative pruning of the face tree for the topaz crystal, where a minor subface had not been deleted; further analyses below will provide other examples.

6.6.4.2 Lamprophyllite crystal analysis

Figure 6.17(a) reproduces figure 6.6(c), and shows the pruned tree of faces found by multiresolution analysis of the lamprophyllite crystal profile. Figure 6.17(b) shows the labels assigned to the faces and figure 6.17(c) makes the hierarchy of the face-tree explicit.

This face-tree, with three levels and many subfaces, represents 18 different maximal consistent face sets. The faces, however, possess certain features which mean that the set of faces which may be considered best on the grounds of symmetry can be selected without producing all 18 separate symmetry analyses. This is possible through consideration of the symmetry properties associated with the more significant faces which are free of subfaces and which are therefore guaranteed to appear in the 'best' face set.

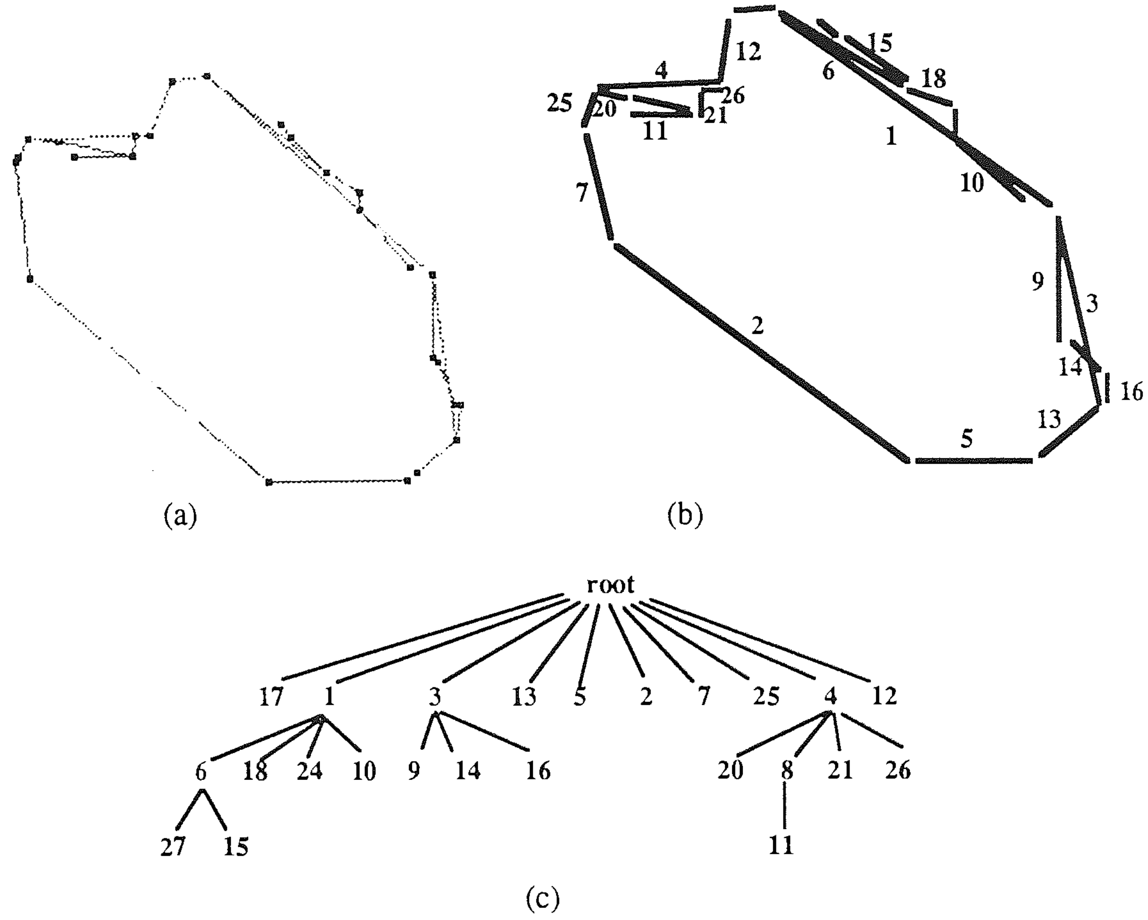


Figure 6.17 Hierarchy of faces for lamprophyllite crystal profile

Table 6.7 shows the master stereogram for this face tree, and provides information on the significance values found for the faces. The most significant face on the profile is face 2. From the face tree it can be seen that this face has no subfaces and is not a subface of another face, and so it must be a member of every maximal face set. A search for faces parallel to this reveals that face 1, the second most significant face in the face tree, is almost exactly parallel.

These two factors, i.e. high significance and good parallelism, seem adequate grounds for assigning face 1 to the 'best' face set. With face 1 assigned, its subfaces can be disregarded, effectively removing its subtree from the face tree. The reduced tree which results represents only 6 maximal consistent face sets.

PD	angle	significance	faces assigned
41	0	49	24
38	187	65	21
36	262	64	17
31	8	129	16
29	302	162	18
27	338	145	14
25	51	162	13
39	203	332	25, 12
35	269	513	17, 4
13	352	476	3
11	174	728	7
40	0	1600	24, 9
9	0	1089	9
7	309	775	6
2	131	5303	2
4	312	1532	1
6	90	3025	5
45	270	1440	26, 20, 11
42	319	1464	27, 15, 10
18	319	315	10
43	319	520	27, 10
33	282	460	20, 8
44	275	235	26, 20

Table 6.7 The stereogram list for the face set of figure 6.17

The procedure can be repeated: the next most significant face is face 5, which also is not a subface and has no subfaces. A search for parallel faces finds that faces 4, 20, 11 and 26 are all approximately parallel to face 5. From the face tree it can be seen that face 4 has two alternative sets of subfaces which could replace it: {20 8 21 26} and {20 11 21 26}. On the grounds of parallelism alone it is not clear whether face 4 is preferable to the second set of subfaces, but the first set can be disregarded. Thus the best maximal face set should contain either face 4 or the subset of faces {20, 11, 21 26}. This reduces the number of possible maximal face sets to 4, namely:

(a) [1 9 14 16 13 5 2 7 25 20 11 21 26 12 17]

(b) [1 3 13 5 2 7 25 20 11 21 26 12 17]

(c) [1 9 14 16 13 5 2 7 25 4 12 17]

(d) [1 3 13 5 2 7 25 4 12 17]

Repeating this process once more, the next most significant face on the profile without subfaces or a parent face is face 7 (face 6 has been removed as part of the face 1 subtree and face 9 is a subface of face 3). Face 7 must be a member of the 'best' maximal face set. A search for parallel faces finds that face 3 is almost exactly parallel (2° from parallel), face 9 is 6° from parallel, face 16 is 14° from parallel and face 14 is 16° from parallel. On the grounds of parallelism, therefore, one would assign face 3 to the 'best' face set, rather than its subfaces 9, 14 and 16. This leaves only two candidates for 'best' face set: sets (b) and (d) above.

This exhausts the list of faces which are both free of subfaces and of relatively good significance. Fuller symmetry analysis of the two remaining candidate face sets may therefore be made.

a) Face set:

{1 9 14 16 13 5 2 7 25 20 11 21 26 12 17}

Parallelisms:

faces	significance
21: 16	65
21: 9	65
7: 9	728
2: 1	1532
5: 26, 20, 11	1440
5: 17	64

No. of mirror planes: 13

Total signif. measure: 40074

3 best mirror planes:

direction	significance
41	7043
0	6623
111	5613

b) Face set:

{1 3 13 5 2 7 25 20 11 21 26 12 17}

Parallelisms:

faces	significance
7: 3	476
2: 1	1532
5: 26, 20, 11	1440
5: 17	64

No. of mirror planes: 11

Total signif. measure: 32450

3 best mirror planes:

direction	significance
41	6157
0	5881
111	5088

c) Face set: {1 9 14 16 13 5 2 7 25 4 12 17}

Parallelisms:

faces	significance
7: 9	728
2: 1	1532
5: 17, 4	513

No. of mirror planes: 11

Total signif. measure: 30820

3 best mirror planes:

direction	significance
41	6634
111	4492
87	4194

d) Face set: {1 3 13 5 2 7 25 4 12 17}

Parallelisms:

faces	significance
7: 3	476
2: 1	1532
5: 17, 4	513

No. of mirror planes: 10

Total signif. measure: 26091

3 best mirror planes:

direction	significance
41	5748
111	4032
132	2986

Table 6.8 Symmetry analyses for 4 likeliest interpretations of figure 6.17

Table 6.8 summarises the results of the symmetry analyses for the final four face sets considered above. From these results face set (b) would appear to have slightly better symmetry properties than set (d), although the differences are not great. Figure 6.18(a) presents the final selection of faces; figure 6.18(b) repeats the original contour for comparison.

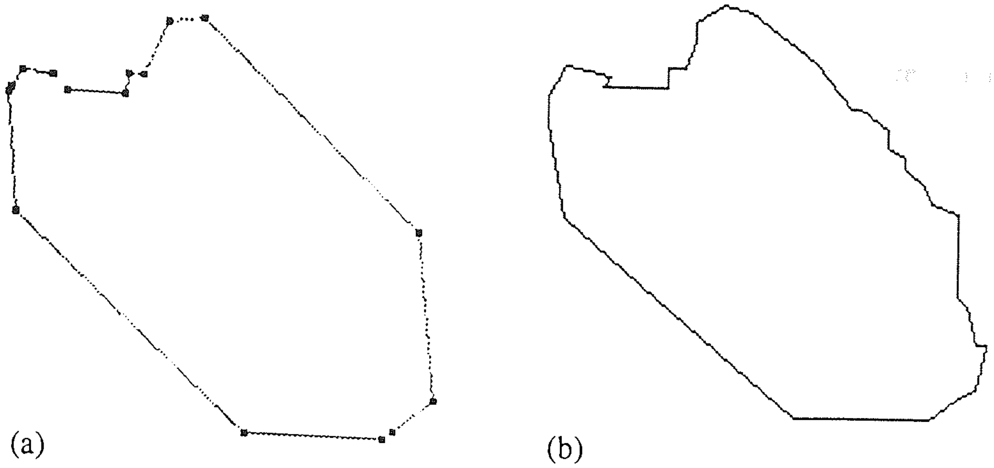


Figure 6.18 Final selection of faces for lamprophyllite crystal

The symmetry analyses for face sets (a) and (c) were also included in table 6.11 since they raise some other issues. It can be seen that sets (a) and (c) yield higher symmetry scores than sets (b) and (d) respectively, although the latter sets were selected above. Sets (b) and (d) were preferred because of the better parallelism between faces 3 and 5 compared to faces 5 and 9. The latter parallelism is, however, rated more highly by the above analysis. This reflects two facts: first, subface 9 has been rated as more significant than face 3 because it is less noisy and, second, the measure of parallelism presented in the table uses only a simple tolerance threshold (10°) to establish parallelism and otherwise does not reflect the degree of prallelism between faces.

There appears to be another important factor which the automated analysis does not consider: what may be termed the 'completeness' or 'simplicity' of the interpretation. Choosing subface 9 in preference to face 3 forces one to accept subfaces 14 and 16 also, and these faces have no obvious part in the symmetry of the figure. Face 3 is parallel to face 5 and so provides a reasonable explanation of that part of the contour; choosing face 9 leaves a portion of the contour 'inadequately explained'. The situation with face 4 and its subfaces is different since subfaces 20, 11 and 26 are all approximately codirectional with face 4, and so choosing the subfaces in preference to face 4 still results in most of that part of the contour lying parallel to a known face.

This example therefore suggests a need for an improved measure of parallelism which reflects the actual angular difference found between the principal directions, in place of the simple tolerance threshold currently used. The example also suggests the need to consider 'completeness' or 'simplicity' when choosing one set of faces in preference to another.

6.6.4.3 Olivine crystal analysis

Figure 6.19(a) reproduces figure 6.4(c), and shows the pruned tree of faces found by multiresolution analysis of the olivine crystal profile. Figure 6.19(b) shows the labels assigned to the faces and figure 6.19(c) makes the hierarchy of the face-tree explicit.

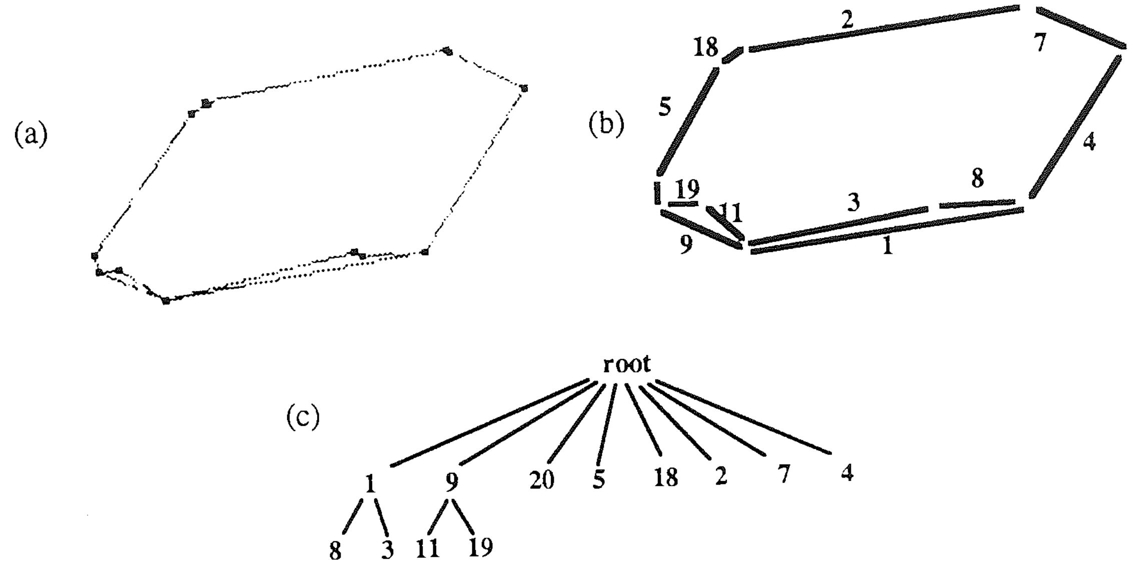


Figure 6.19 Hierarchy of faces for olivine crystal profile

The face tree represents four possible maximal face sets, namely:

(a) [8 3 11 19 20 5 18 2 7 4]

(b) [8 3 9 20 5 18 2 7 4]

(c) [1 11 19 20 5 18 2 7 4]

(d) [1 9 20 5 18 2 7 4]]

PD	angle	significance	faces assigned
23	247	29	18
21	172	50	20
19	90	64	19
13	129	274	11
11	301	341	7
9	221	1224	5
7	82	1188	3
2	264	3104	2
4	37	1624	4
18	86	1348	19, 1
6	86	1090	1
17	92	450	19, 8
14	92	218	8
16	119	178	9

Table 6.9 The stereogram list for the face set of figure 6.19

The master stereogram for the tree of faces is shown in table 6.9. Table 6.10 lists all the parallelisms found, using a tolerance of 10°, expressed in terms of the principal direction (PD) numbers and not the face labels.

Label	Significance	PD1	PD2
9	274	13	11
8	450	17	2
7	218	14	2
6	178	16	11
5	64	19	2
4	1348	18	2
3	1090	6	2
2	1188	7	2
1	1222	4	9

Table 6.10 List of parallelisms derived from master stereogram list

Table 6.11 summarises the results of the symmetry analyses of the four maximal face sets. The parallelisms found are indicated by the labels used in table 6.10. The four analyses of table 6.11 show very similar results: the three best mirror planes found in all cases are virtually identical, and the parallelisms and significance scores are also similar.

a) Face set: {8 3 11 19 20 5 18 2 7 4}

Parallelisms: 1, 2, 8, 9

No. of mirror planes: 9

Total signif. measure: 20469

3 best mirror planes:

direction	significance
173	5189
129	4382
151	3372

b) Face set: {8 3 9 20 5 18 2 7 4}

Parallelisms: 1, 2, 6, 7

No. of mirror planes: 8

Total signif. measure: 17929

3 best mirror planes:

direction	significance
173	4628
129	4287
151	3139

c) Face set: {1 11 19 20 5 18 2 7 4}

Parallelisms: 1, 4, 9

No. of mirror planes: 9

Total signif. measure: 18580

3 best mirror planes:

direction	significance
174	4739
129	4332
151	2956

d) Face set: {1 9 20 5 18 2 7 4}

Parallelisms: 1, 3, 6

No. of mirror planes: 8

Total signif. measure: 16054

3 best mirror planes:

direction	significance
129	4237
175	3899
151	2824

Table 6.11 Symmetry analyses for 4 interpretations of figure 6.19

The results presented in table 6.11 show that the most significant single mirror plane and highest sum of mirror plane significances are found for face set (a), i.e. the face set which opts for all the subfaces rather than the parent faces. This set also contains four pairs of parallel faces: faces 2 and 3, 4 and 5, 7 and 11, and face 2 and the codirectional pair 8 and 19.

It is not certain, however, that face set (a) is the one which would be selected as 'best' by a human observer. A face set which is at least as likely is set (d), i.e. the set which opts for all the parent faces rather than the subfaces, and set (c) could also be argued for. As with the lamprophyllite example, introspection suggests two deciding factors: parallelism and simplicity / completeness. Neglecting the two very minor faces 18 and 20, face set (d) consists of six major faces which constitute three parallel pairs of faces: 1 and 2, 4 and 5, 7 and 9. The face set therefore represents a simple interpretation with strong symmetry properties.

The simple strategy for face selection used above, i.e. seeking faces parallel to significant, guaranteed members of the 'best' face set, is again applicable. In this face tree faces 2, 4, 5 and 7 have no subfaces or parent faces and so are certain members of any maximal face set. In addition, faces 2, 4 and 5, and face 7 to a lesser extent, have relatively good significance values. Starting with the most significant face of this type, i.e. face 2, a parallel face can be sought. Faces 3 and 1 are both 2° from being exactly parallel; face 8 is 8° off parallel. Thus both choices provide simple 'explanation' of the lower side of the figure, with good symmetry properties. The choice of face 1 may be preferred on the further grounds of simplicity.

Faces 4 and 5 are parallel to each other and form no parallel pairs with faces from the undecided portion of the contour. Face 7 has, however, been found parallel to face 9 (2° off); face 11 is 8° from parallel, and its sibling subface 19 is codirectional with face 1 and so is parallel to face 2. Thus again both choices have good symmetry properties. On the grounds of simplicity alone, face set (d) would be preferable, but clearly either decision could be defended. Figure 6.20(a) shows the faces of set (d); figure 6.20(b) shows the original contour.

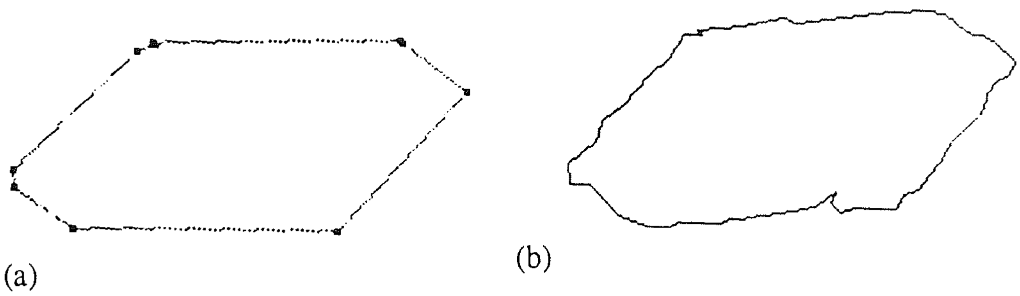


Figure 6.20 Olivine crystal face set (d) and original profile

6.6.4.4 Augite crystal analysis

Figure 6.21(a) reproduces figure 6.5(c), and shows the pruned tree of faces found by multiresolution analysis of the augite crystal profile. Figure 6.21(b) shows the labels assigned to the faces and figure 6.21(c) makes the hierarchy of the face-tree explicit.

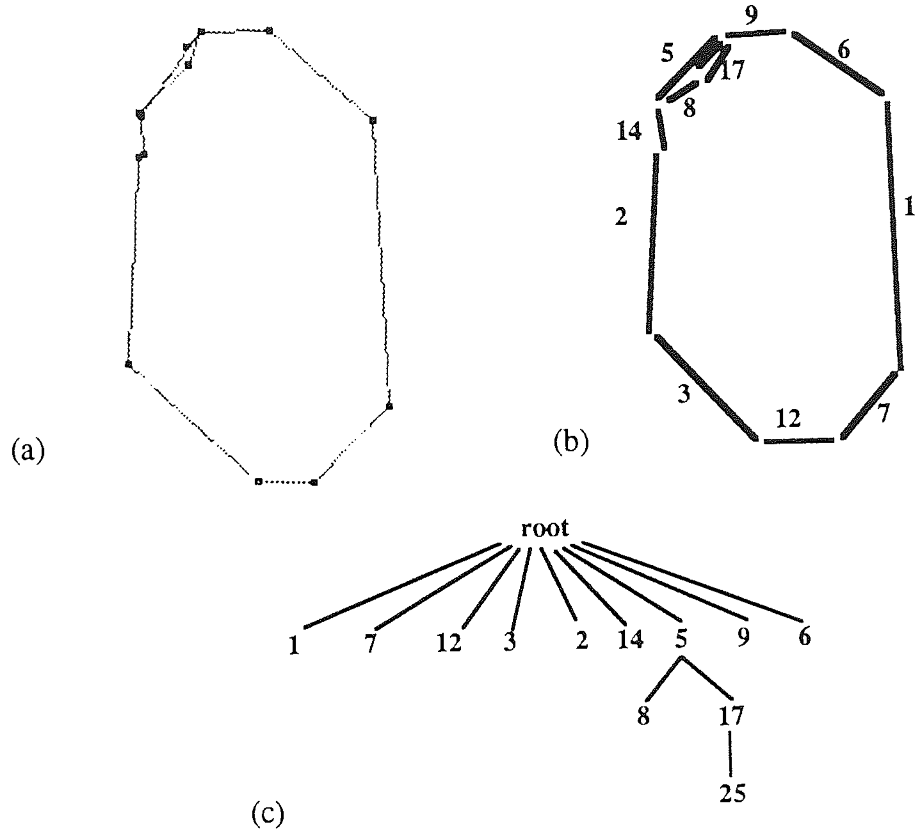


Figure 6.21 Hierarchy of faces for augite crystal profile

The face tree represents three possible maximal face sets, namely:

(a) [1 7 12 3 2 14 8 25 9 6]

(b) [1 7 12 3 2 14 8 17 9 6]

(c) [1 7 12 3 2 14 5 9 6]

PD	angle	significance	faces assigned
22	226	513	25, 8
18	226	253	8
16	216	214	5
14	90	484	12
12	268	730	9
10	43	1739	7
19	182	1899	14, 2
6	310	2837	6
4	131	2335	3
2	356	2415	1
21	201	56	17

Table 6.12 The master stereogram list for the face-tree of figure 6.21

The master stereogram for this tree of faces is shown in table 6.12. There are principal directions for faces 8 and 25 together, and for face 8 separately, since face 25 may be replaced by its subface 17. Faces 14 and 2 have been found to be co-directional and are assigned to the same principal direction.

Table 6.13 lists all the parallelisms found, using a tolerance of 10°, expressed in terms of the principal direction (PD) numbers of the master stereogram and not the face labels.

Label	Significance	PD1	PD2
6	253	18	10
5	214	16	10
4	1899	19	2
3	2335	4	6
2	484	14	12
1	513	10	22

Table 6.13 List of parallelisms derived from master stereogram list

This example shows several relatively significant faces with no subfaces or parent faces and which are therefore guaranteed members of the 'best' face set: faces 1, 2, 3, 6 and 7. Of these, only face 7 is parallel to any faces which are not present in all three maximal face sets. From tables 6.12 and 6.13 it can be seen that face 7 (PD 10) is parallel to face 5 (PD 16), to face 8 (PD 18), and to face 25 (PD 22). This therefore suggests that maximal face sets (a) and (c) are preferable to set (b).

a) Face set: {1 7 12 3 2 14 8 25 9 6}

Parallelisms: 1, 2, 3, 4

No. of mirror planes: 8

Total signif. measure: 49180

3 best mirror planes:

direction	significance
40	10472
89	9162
176	7427

b) Face set: {1 7 12 3 2 14 8 17 9 6}

Parallelisms: 2, 3, 4, 6

No. of mirror planes: 9

Total signif. measure: 47318

3 best mirror planes:

direction	significance
40	10211
89	8958
176	7166

c) Face set: {1 7 12 3 2 14 5 9 6}

Parallelisms: 2, 3, 4, 5

No. of mirror planes: 8

Total signif. measure: 46300

3 best mirror planes:

direction	significance
40	10172
89	8862
176	7127

Table 6.14 Symmetry analyses for 3 interpretations of figure 6.20

On the grounds of simplicity alone, face 5 may be preferred to its subfaces 8 and 25. A consideration of the symmetry analyses for these two face sets would suggest that face set (a) is preferable, but there is little to choose between them.

Table 6.14 summarises the symmetry analyses of all three maximal face sets. It may be noted that face set (b), which was rejected because of the lack of parallelism, yields one more mirror plane than the other face sets (since face 25 introduces a new direction into the stereogram) and produces a higher total significance value than face set (c). This supports the observations made above that introducing a face which adds a new direction to the stereogram is likely to produce new mirror planes, increasing the total significance value, and so the total significance value is not a good indication of the best face set.

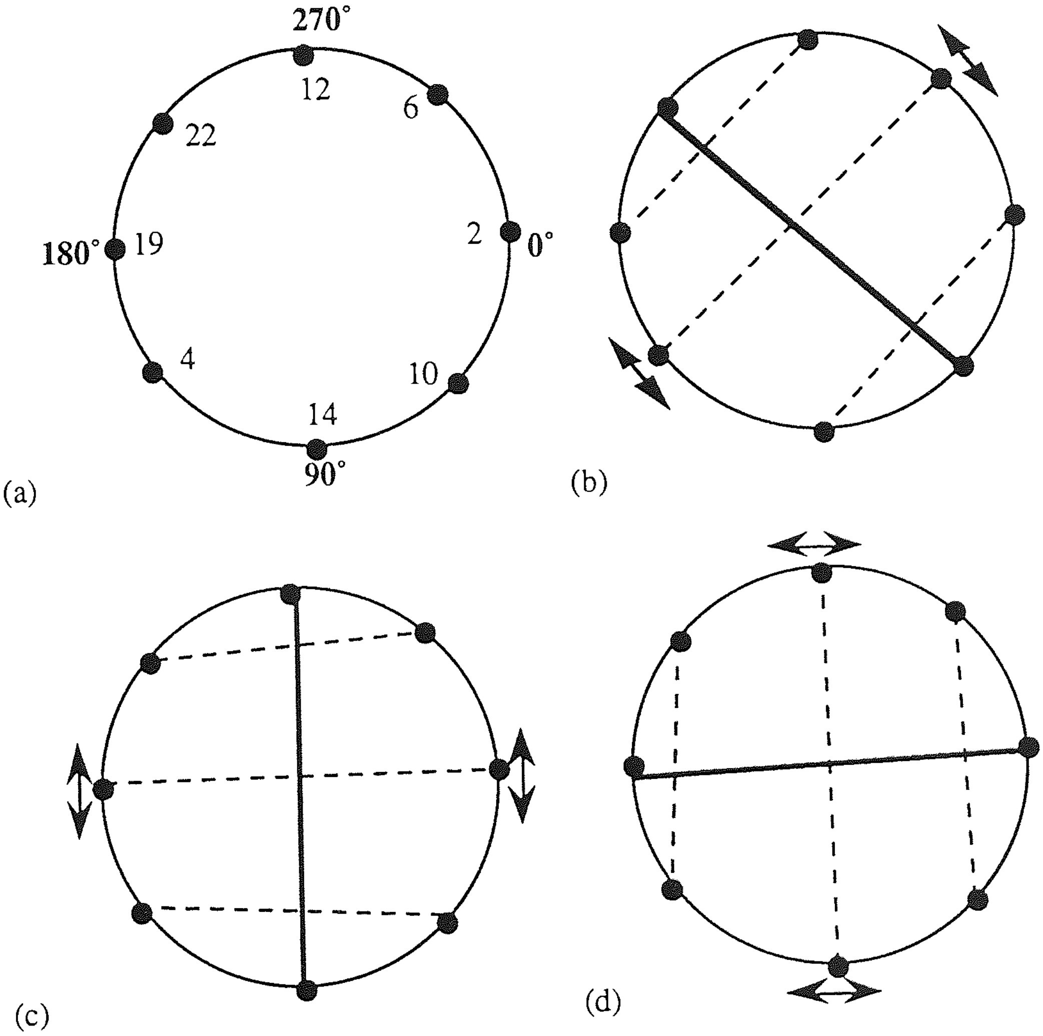


Figure 6.22 Symmetry analysis of augite crystal

All three face sets yield the same three most significant mirror plane directions: at 40°, 89° and 176°. Figure 6.22 represents the symmetry analysis for maximal face set (a). Figure 6.22(a) shows the reduced stereogram with the principal direction labels indicated. Figure 6.22(b), (c) and (d) show the three most significant mirror planes found in the analysis, displayed using the same conventions as in figure 6.7.

Figure 6.23(a) shows face set (a); figure 6.23(b) repeats the original augite crystal profile.

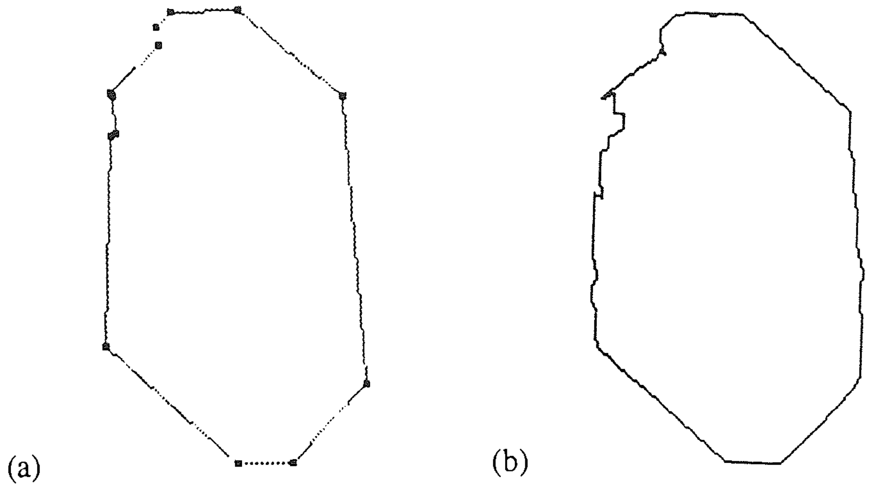


Figure 6.23 Augite crystal face set (a) and original profile

6.7 Further development and conclusions

The work described above represents the first steps in a large and difficult problem area. In this section the directions of further improvements and development will be discussed, the contributions of this work will then be summarised and some conclusions drawn.

6.7.1 Further work

The algorithms described above were developed to investigate the usefulness of some new ideas in symmetry analysis, and are not proposed as final or complete solutions to the problems outlined above [6.1]. The implementations may thus be viewed as simple, fast 'prototypes'. Some of the limitations of the algorithms have been exposed in the example analyses above.

Probably the feature of the algorithms most clearly in need of improvement is the use of a fixed tolerance when finding parallel faces. The current implementation uses a fixed tolerance threshold to decide whether two faces are parallel: if their principal directions differ by 180° to within a set tolerance (10° was used in the above examples) then the two faces are classed as parallel and a measure of the significance of the parallelism is made. If the principal directions do not differ by 180° to within the set tolerance, even if they are only 0.1° beyond the set limit, then the faces are not considered parallel. This clearly is unsatisfactory.

A simple improvement would be to introduce what may be termed an 'attenuation factor' Δ, a real number in the range [0,1], which reflects the departure of the actual angular difference from the ideal value of 180°. The Δ factor would then be used to multiply (and so reduce) the significance value derived for the parallelism. For two faces exactly 180° apart the Δ factor would be 1; the factor would fall to zero with increasing departure from 180°.

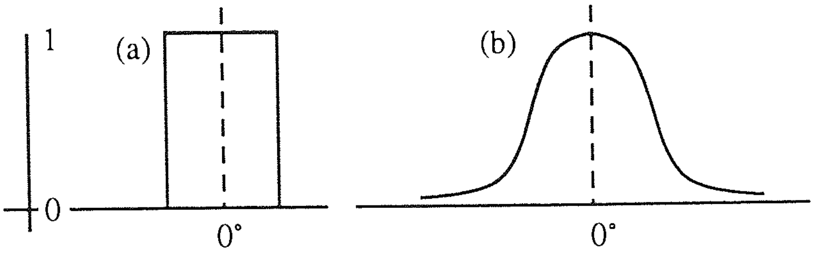


Figure 6.24 Two attenuation functions

Figure 6.24(b) sketches one form of the attenuation factor which might be appropriate, i.e. a function similar in profile to the normal distribution. Experimentation would be required to establish a suitable profile. Figure 6.24(a) shows the simple step function which, when used as an attenuation factor, is equivalent to the fixed threshold scheme currently employed.

It may also be possible to use a similar technique when clustering the face normal directions to find the principal directions of the reduced stereogram or master stereogram, or when clustering the hypothesised mirror planes (hmps) when seeking significant mirror planes, since the current implementations use fixed tolerance thresholds (of 7° and 10° respectively). It may be possible to make the clusters (i.e. the member-faces sets associated with each principal direction of the stereogram and the 'support' lists associated with each mirror plane) into fuzzy sets [Zadeh 1965]. The membership function used to define the degree to which an element belongs to a fuzzy set would be similar to the attenuation function described above. The application of fuzzy set theory is therefore a suitable area for further research.

Other areas for further work have been mentioned above [6.4.1]: to review the significance measures used for faces and principal directions and to review the rules for deriving significance measures for symmetry features from the significance values of the principal directions involved. One point for consideration has also been identified: is it possible to formalise and quantify the 'completeness' or 'simplicity' of a face set, i.e. the extent to which it gives every part of the contour some satisfactory 'explanation' [6.6.4.2].

The matter of how mineral cleavage information can be brought into the symmetry analysis must also be considered. As has been described [2.2.5; 3.5.2.3], cleavage is very important in establishing the symmetry of a mineral. Without such information it may be the case that the symmetry descriptions derived by these methods will prove insufficiently powerful to characterise the crystallographic symmetry classes to adequately support mineral identification.

6.7.2 Summary and conclusions

This chapter has investigated how some concepts of crystallographic symmetry analysis can be implemented and used. The analyses produced are different from those derived by 'conventional' symmetry analysis algorithms [6.2], being concerned with face orientations, regardless of the ordering of the faces on the contour. This work presents several novel contributions to the analysis of two-dimensional form.

A scheme for representing symmetry information symbolically has been devised and several new algorithms for symmetry analysis have been presented. The key intermediate data structure termed the 'reduced stereogram' has been introduced. Techniques for stereogram analysis to find parallelisms and possible axes of mirror symmetry have been developed.

The analysis techniques have been extended to support the more efficient analysis of multiple data sets drawn from the same face tree, taking into account restrictions on which sets of faces can represent valid interpretations of a contour. The idea of maximal consistent face sets was introduced and an algorithm for deriving the list of all such face sets from a face tree was presented. Similarly, the 'master stereogram' was introduced, and its construction described.

The modified techniques for the analysis of face trees directly support the use of symmetry analysis in selecting the 'best' face set for a crystal profile. The second use of symmetry analysis in the assignment of characteristic boundaries, i.e. in deciding to which of two crystals a straight interface should be assigned, has not been explicitly addressed. The techniques for finding parallelisms and mirror planes will still be applicable; the significance measures are also expected to be adequate for indicating the better alternative.

The work of this chapter is original, apart from the use made of the simple clustering algorithm first applied to shape analysis by Tanimoto [1980]. The approach to analysis presented here was developed independently, however, and the algorithms which have been developed go beyond the simple shape measures presented by Tanimoto in the novelty and difficulty of the problems tackled and the complexity of their implementations.

The work of this chapter forms a natural continuation to that of the previous chapter. As described above, the development of the analysis techniques is not fully complete, and directions for further work have been discussed. The results of the investigations reported above are, however, considered sufficiently promising to indicate that useful symmetry information can be derived from a description of possible crystal faces, such as that produced by the multiresolution analysis described in the previous chapter, and that the techniques presented above show the way forward to achieving this.

The techniques presented above may be found to be valuable in applications of image analysis and understanding in other problem domains beyond petrography and microscopy, where objects of interest are represented by image regions with straight boundary segments and have symmetric form. One possibility is the use of such symmetries for grouping linear features in aerial photographs or satellite images, thus aiding the recognition or better delineation of buildings, fields and other man-made artefacts; it may also be possible to use these new definitions of symmetry to aid object recognition or classification in bin-picking and similar industrial machine vision applications.

CHAPTER 7

System Overview

7.1 Introduction

Chapter 3 [3.6] outlined a scheme for the analysis of polycrystalline igneous rock thin section micrographs which should be capable of assigning the characteristic boundaries of the crystals and producing useful, geologically relevant descriptions of the crystal form. Figure 7.1 recalls figure 3.11, showing the component modules which have been identified as necessary for a successful analysis.

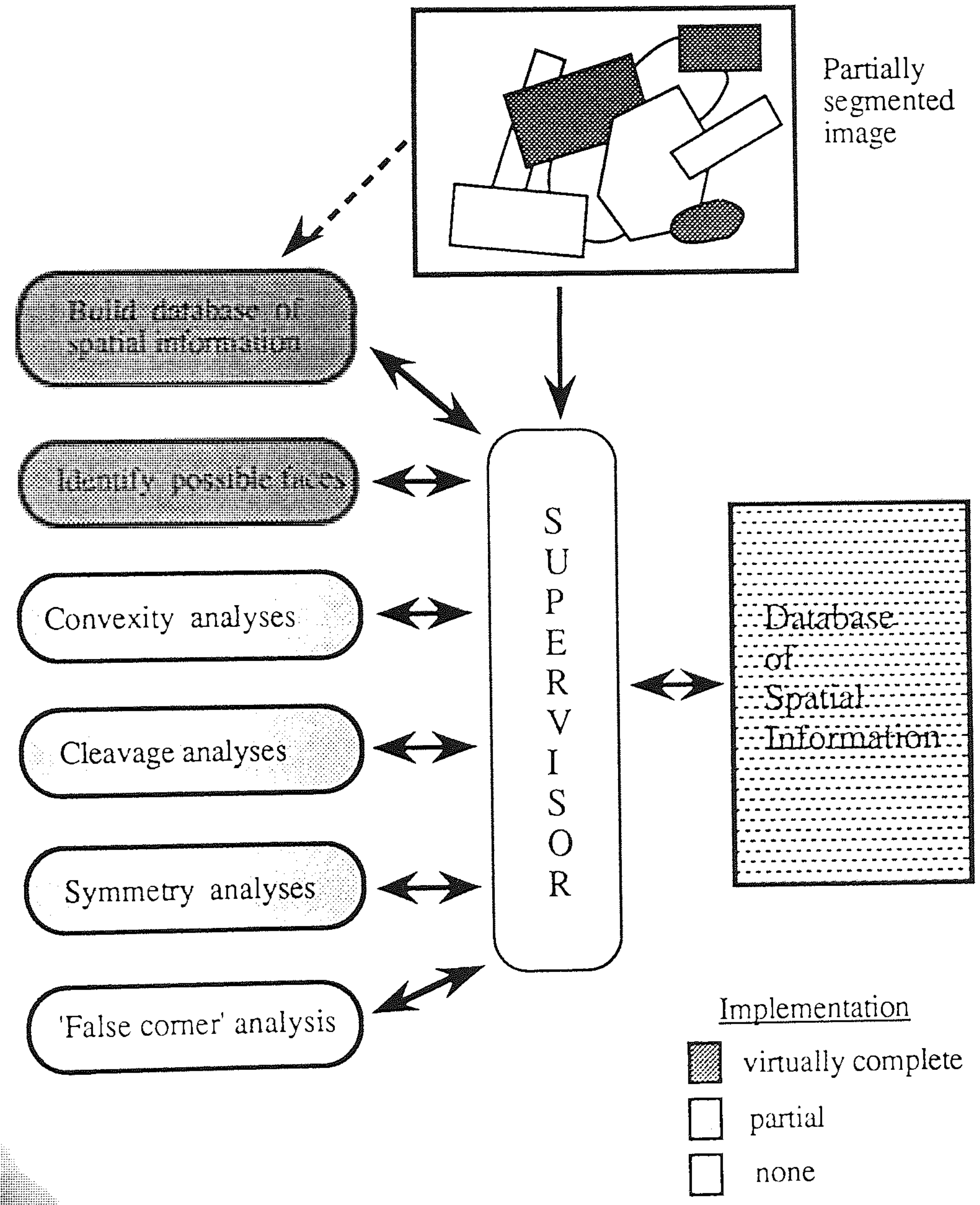


Figure 7.1 Petrographic image analysis system outline

The figure shows a suggested control structure where a supervisor module provides an interface between the spatial information database and the analysis modules. The diagram has been shaded to indicate the approximate state of development of the various modules; details are given below. The implementation of some the key analysis tasks has been considered in detail in the previous three chapters. Other components of the analysis have also been investigated and have been reported elsewhere [Thomson & Sokolowska 1988; Thomson & Claridge 1989; Thomson 1991].

In this chapter the current state of development of the petrographic image analysis system will be reviewed and further work required will be discussed. In particular, a possible control strategy for the various analyses is outlined, indicating how the image analysis system may be implemented.

7.2 Current implementation status

7.2.1 Building the SPIN database

As was reported in chapter 4, the 'Spinster' algorithm which extracts the required structural and topological information from a partially segmented image and constructs a symbolic description in the form of the SPIN spatial information database has been developed and implemented. An extension to the algorithm to cater for intra-region 'cracks' in the image has been proposed but not yet realised. The Spinster algorithm therefore represents a virtually complete implementation of the module required to build the database of spatial information (figure 7.1).

The SPIN database produced by the Spinster algorithm represents only the basic spatial information such as region-region adjacency, region contour chaincodes and vertex-boundary incidence. Further information derived by the other analysis modules will have to be added to the database. Representations for some of this new information in the database have already been devised, e.g. the face tree [6.1.1] and symmetries [6.4]; a full representation scheme has not, however, been completed. The storage of these items of information in the database would be handled by the supervisor module.

7.2.2 Identifying possible faces

The task of finding faces on a crystal profile has been found to require both local analysis of contour straightness and non-local analysis of symmetry [5.1.1]. The problem has been tackled in two stages. First, a multiscale analysis produces a hierarchical description of the linear and near-linear structure on the contour, in the form of the face tree. Next, if ambiguity about the strucure remains, symmetry analysis is performed to select the best set of faces from the tree structure. The module for identifying possible faces is required to perform only the first stage of this analysis, i.e. to produce the face tree and descriptions of the faces present.

The implementation of this module has, in effect, been achieved. Chapter 5 described in detail a multiscale analysis algorithm which found possible faces on a contour, constrained to form a tree structure. Chapter 6 included a description of how the face tree data structure could be represented in the SPIN database [6.1.1], described the implementation of an algorithm for its construction, and also showed how the tree could be pruned of unnecessary faces [6.1.2], improving the efficiency of subsequent analyses.

Further improvements of the multiscale analysis have been outlined [5.5.1], although the algorithm appears to work acceptably well. Table 7.1 repeats table 5.13, summarising its performance over ten test profiles (eight real data and two synthetic examples). The time complexity of the algorithm was estimated at $O(N^2)$, where N is the contour length measured in chaincode chainlets, with the space requirements $O(N\log_2 N)$ [5.4.4].

	False Negatives		False Positives	
class	Major	Minor	Major	Minor
Partial miss	2	9	0	12
Full miss	1	49	0	49
Max possible error count	84	194	82	195

Table 7.1 Multiscale analysis performance summary for ten test examples

7.2.3 Convexity analysis

Chapter 3 demonstrated that convexity is a key property in understanding the assignment of characteristic boundaries in polycrystalline section images. Procedures were proposed in which the convex hull of a profile was formed and intersected with a neighbouring profile in order to decide relative crystal 'precedence' [3.5.2.1; 3.5.3]. It was also shown that a comparison between a profile and its own convex hull may give a useful indication of its shape and identify embayments which could be examined further to see if they were imposed or the result of non-convex growth.

The implementation of such analyses has been considered. An algorithm for finding the convex hull of a set of points in two dimensions was implemented. The algorithm in its basic form is due to Jarvis [1973; Akl 1979]. Useful variants were also found, notably a modification for constructing the convex hull of a single closed contour and, at the same time, producing descriptions of the concavities or 'pockets' found on the contour. This version is appropriate for the analysis of single crystal profiles. The use of the convex hull algorithm in interface analysis was also considered, and guidelines for its use were set down. Algorithms were also found for two other tasks required in using convexity information to derive information about relative crystal 'precedence': finding the intersection of two regions and finding the areas of regions.

The investigations suggested that the implementation of the relevant algorithms would be relatively straightforward, and require few novel additions or amendments to standard techniques. Consequently, the study of convexity has not been presented in detail in this thesis; a separate report has been prepared [Thomson 1991].

7.2.4 Cleavage analysis

The importance of crystal cleavage in thin section analysis was described in chapter 3: cleavage planes reveal some of the internal structure of a crystal and can therefore be used to recognise characteristic faces via symmetry relations. Cleavage can also be an important clue in mineral identification.

Some investigations have been made into the problems of recognising cleavage cracks amongst other textural features, describing their quality of development and measuring their orientations. No completely satisfactory solution to the problems of cleavage analysis was achieved, although some interesting possibilities were raised.

The Hough transform [Duda & Hart 1972; Illingworth & Kittler 1988] was investigated as mechanism for cleavage identification. This transform is a popular and efficient method for detecting parameterised entities (e.g. straight lines, ellipses) in noise, and apparently well suited to detecting significant alignments in images. It was found, however, that additional processing was required to achieve this in practice; the use of a parameter space convolution technique, due to Boyce *et al.* [1987], was proposed. Projections were used to identify the relevant features in the transform space; directional filtering was used to isolate the cleavages. Further tests on these methods are required. This work has been presented and published elsewhere [Thomson & Sokolowska 1988].

(It should be noted that cleavage analysis requires access to the original image data and the partially segmented image serves only to provide 'masks' to isolate the appropriate areas for analysis. Figure 7.1 does not show the use of the original image data.)

7.2.5 Symmetry analysis

Two uses of symmetry analysis in correctly assigning characteristic boundaries have been recognised [6.1]. Symmetry information may be required in selecting one final set of faces for a crystal profile from the various options presented by the face tree. It was also shown that symmetry analysis may be required in deciding to which of two crystals a straight interface should be assigned (see figure 3.8). A third rôle for symmetry analysis has also been recognised: building descriptions of crystal symmetry which can be used in identifying the mineral. Such analysis is not required for the correct assignment of characteristic boundaries.

The symmetry analyses module will therefore contain routines supporting two different types of analysis. Chapter 6 reported considerable progress towards the implementation of the required data structures and algorithms. The implementations achieved have been described as 'prototypes' [6.7.1]; suggestions for further improvements have been given.

A scheme for the representation of symmetries in the SPIN database was devised; algorithms for finding parallelisms and mirror symmetries were developed and implemented. The techniques introduced were extended to support the more efficient analysis of face trees; the techniques are also expected to be applicable in the assignment of straight interfaces.

7.2.6 False corner analysis

'False corner' analysis, also termed the 'T-junction' test [2.5.2.3], concerns the straightness or smoothness of crystal profile boundaries at vertices, i.e. the points where three or more crystals meet. This analysis has not been implemented. Implementation of this module is expected to be relatively simple and straightforward, however, requiring little more than an examination of the spatial information database.

The SPIN database when initially created provides information regarding vertex-region boundary incidence, and other analyses will provide information about faces on region boundaries. It should therefore be simple to inspect the database to see if a vertex lies within a face on one of the crystal boundaries. If so, then a T-junction has been found and the region showing the smooth continuation of contour at the vertex has the characteristic boundary. In cases where no region has a face at the vertex an inconclusive verdict will be returned.

7.3 System control: the supervisor

The supervisor module shown in figure 7.1, representing an overall control mechanism for the analysis, has not yet been implemented. This module calls upon the various analysis modules as it tries to construct reliable descriptions of the crystal shapes and inter-crystal relations. In particular, it aims to solve the problem of assignment of characteristic boundaries.

It is intended that the central control routines will be written in POP-11. This language facilitates the implementation of 'higher level' control tasks and database handling, and also allows access to procedures in other languages, such as C. This is important since some image processing procedures already exist or are more efficient in this language.

The following is an outline of a basic control strategy for coordinating the analysis activities. This strategy takes into account the relative usefulness of the analyses [3.5.4.1] and considerations of computational efficiency. Thus, for example, symmetry analysis is only performed for purposes of boundary assignment if convexity has provided no definite answer, and the T-junction test is applied last of all, and only if previous analyses have been unable to provide the required information. A test is not carried out if a more 'powerful' analysis could overturn its findings later.

7.3.1 Control strategy

The first step in the analysis of the image is the construction of the SPIN spatial information database. As the analysis proceeds this database will be both interrogated by the supervisor and updated through the addition of the new information which results from the various analysis. Thus the database in its final form will also contain information about faces on contours, symmetry descriptions, and the necessary information regarding the attribution of characteristic boundaries.

The first set of inferences on boundary ownership requires only a simple inspection of the database: the Spinster algorithm for extracting spatial information recognises when a region or group of regions is completely surrounded by another single region and stores the fact in the database. If such information is present it provides unequivocal evidence for the ownership of the faces concerned, since it represents crystal inclusion. The boundary assignment information is stored in the database.

Contour analyses are then made. Possible crystal faces are identified on each crystal profile; the face tree produced for each profile is stored in the database. The whole boundary is treated as a characteristic boundary for this analysis; the decisions on which faces found on a profile are really characteristic are made later. The second aim of the contour analysis is data compression, providing a polygonal approximation for each contour which improves the efficiency of the convexity analyses.

When this analysis is complete for each crystal profile convexity analyses can be made, testing each pair of adjacent crystal regions for partial enclosure. Interfaces for pairs of regions where one wholly surrounds the other need not be considered, since the enclosure has already been established. As previously described, the SPIN database distinguishes between the two types of adjacency, aiding the control program. Again, the inferences drawn concerning boundary ownership are stored in the database.

If at this point there is still doubt as to the ownership of some boundaries symmetry analysis can be invoked to try to settle the matter. If several alternative sets of faces are available for region contours the symmetry analysis can first be used to choose the optimal set. Cleavage analysis of the regions concerned may also be invoked, providing more information about the crystal structure which can be used in the symmetry analysis. Faces of uncertain ownership would be examined to see how they fit into the symmetry schemes of the two crystals to which they could possibly belong, and assigned to the crystal which offers the better 'explanation' for this face.

Finally, if doubt still remains as to the ownership of faces the false corner analysis is tried. Any ambiguities remaining at this stage are reported to the user. It is likely that the efficiency of the analysis will be improved by examining regions and faces in sequences working outwards from 'islands of certainty' - i.e. boundaries whose ownership has been unambiguously established by earlier passes.

The analysis for boundary assignment is now complete. The analyses may be continued, however, if required to provide further descriptive information. Cleavage and symmetry analyses can be finalised for regions, providing information for mineral description and identification. Descriptions of the quality of development of crystal faces can also be derived from information extracted during processing. In particular the proportion of a crystal perimeter made up of characteristic faces, and the straightness of these faces can be found.

CHAPTER 8

Summary and Conclusions

8.1 Overview

This thesis has considered the use of image analysis analysis in petrography and investigated the possibilities for advancing existing techniques by introducing new feature extraction and analysis capabilities. The aim of such analyses is the construction of relevant, useful descriptions of a polycrystalline rock section which cannot be enumerated in advance, and not merely the recognition of the input as one of a pre-defined set of prototypes. Thus petrographic image analysis has been considered as an image understanding problem.

The principal contributions of this study may be considered in three groups: recognising a key problem in petrographic image analysis and establishing requirements for an effective solution, developing and implementing original image analysis techniques, and establishing useful links between this work and more general image analysis problems.

This thesis has identified the fundamental problem of assigning the ownership of characteristic boundaries in polycrystalline rock sections. The problem has been described, explained and analysed. The image features which are key to its solution have been identified; clear analysis rules have been presented. The value of these rules is clear: they indicate the image analysis tasks which must be tackled if characteristic boundary assignment is eventually to be automated.

A petrographic image analysis system based on these principles which addresses the problem of boundary assignment has been outlined. The development of component modules for this system has produced several new techniques, data structures and algorithms for image analysis. The most notable contributions are a new multiscale algorithm for contour analysis, and new symmetry analysis techniques. The analysis modules contribute both to the assignment of characteristic boundaries and to the construction of geologically useful descriptions of crystal form and inter-crystal relations. A chapter by chapter review below gives a more detailed assessment of the work.

Analogies have been established between petrographic image analysis and problems of computer vision where the task is the understanding of two-dimensional images of three-dimensional scenes. In particular, strong parallels have been found between the individual principles which have been derived for establishing the ownership of crystal boundaries and principles of perceptual organisation and figure-ground discrimination. This linkage has been considered only briefly in this thesis; a more detailed treatment has been published elsewhere [Thomson & Claridge 1989]. Because such parallels exist, the petrographic domain provides a suitable area in which to study perceptual processes, and one where advances of both theoretical and practical value may be achieved. These matters will be considered in greater detail below, in the context of future work.

8.2 Thesis review

Chapter 2 presented background information on petrography, and a 'scene-setting' review of image analysis in petrography. This established the importance of quantitative methods in petrography and examined how image analysis is used to automate such techniques. The extension of current petrographic image analysis capabilities was discussed, leading to preliminary proposals for research into the provision of more useful descriptions of crystal form and inter-crystal relations.

Chapter 3 initiated investigations in this direction through a consideration of the fabrics of polycrystalline rocks. Petrological information was given to explain the origin of certain features seen in rock sections, and appropriate terms for their description were discussed. The fundamental problem of boundary assignment was identified. An original study of how this problem is solved by human observers identified image features which reveal boundary ownership. Analysis rules were presented and their relative power assessed. The key precedence indicator, apart from complete inclusion, was found to be convexity; parallelism of faces or, more generally, good symmetry was found to be less important; 'false corners' were found to be the weakest indicators.

These rules clarify the nature of the image analysis tasks which must be tackled if the boundary assignment problem is eventually to be automated. The design of a novel image analysis scheme based on these principles was outlined. The implementations of key modules of this system were considered in detail in subsequent chapters.

Chapter 4 considered the problem of extracting and organising the low-level spatial information describing a partially segmented rock section image. A database of spatial information was created which could serve as the core of the petrographic image analysis scheme proposed in chapter 3. Initially supplying the basic data for such an analysis, the database would be interrogated by a supervisor module in order to monitor the state of the analysis and to supply input data to the various individual analysis routines, and updated through the addition of the new information which results.

Although the algorithms and data structures were designed to support the requirements of petrographic image analysis, the information being handled is fundamental to many image analysis tasks. The techniques described in this chapter may therefore be useful in other image analysis applications, such as histology or remote sensing.

Chapter 5 addressed the fundamental problem of identifying possible crystal faces on a profile. Requirements for useful crystal profile analysis were identified. First, the analysis must operate at a range of resolutions and avoid the use of arbitrary error thresholds. Second, some mechanism should be provided to support the use of non-local shape information such as symmetry or collinearity. Thus, when ambiguity exists regarding the best set of faces on the contour, the additional non-local information can be used in making the final decision. This would provide an implementation of a simple example of the Prägnanz principle: the final analysis represents the 'best' figure. Such an approach has not previously been developed in practical schemes for contour analysis.

A review of contour analysis algorithms was made: no algorithm was found which provided a complete solution to the problems of crystal profile analysis. A new algorithm was therefore devised. The face-tree data structure was created and proposed as a succinct representation of the significant linear structure on a contour in a convenient form for further analysis. A multiscale analysis algorithm which produces a face-tree representation of a contour was then presented in detail.

The algorithm achieves its analysis via a novel intermediate representation of the contour which quantitatively describes the 'corners' and 'sides' found over a range of resolutions. Original procedures were presented for analysing the intermediate representation, using density slicing and a 'greedy' algorithm to refine the set of suggested faces, and for enforcing a strict hierarchy of faces and subfaces. One notable and apparently unique feature of the overall algorithm is its ability to cope with variation in optimal resolution around the contour.

A new scheme for evaluating the effectiveness of the algorithm was developed, extending a current method used to analyse the performance of simple classification algorithms. The evaluation supported the view that the algorithm performs well. The causes of the minor failings which were found have been identified and their correction discussed.

This chapter has therefore made theoretical and practical contributions to two-dimensional shape analysis, introducing new ideas and new algorithms. Since the problem of finding significant linear structure on curves or contours is not unique to petrography, and can be a key step in image understanding, this algorithm may be found valuable in other applications.

Chapter 6 investigated how some concepts of crystallographic symmetry analysis can be implemented and used. The analyses produced are different from those derived by 'conventional' symmetry analysis algorithms, being concerned with patterns discernible in the orientations of crystal faces (and cleavages), irrespective of the ordering of the faces on the contour. The analyses operated on the face-trees produced by the multiscale analysis presented in chapter 5.

This work made several novel contributions to the analysis of two-dimensional form. Schemes for the symbolic representation of symmetry information were devised and new analysis algorithms were presented. The key data structure termed the 'reduced stereogram' was introduced. Algorithms for stereogram construction and stereogram analysis were developed, supporting the identification of parallelisms and possible mirror symmetries.

The techniques were then extended to support the more efficient analysis of multiple data sets drawn from the same face tree. The idea of the maximal consistent face set was introduced and an algorithm for deriving the list of all such face sets from a face tree was presented. Similarly, the 'master stereogram' was introduced, and its construction described.

The techniques have been implemented and their operation on a series of real data examples presented. Directions of further development have been discussed.

Chapter 7 reviewed the state of implementation achieved for the petrographic image analysis system outlined in chapter 3. The constituent analysis modules were each considered, including notes on work not reported in this thesis and published elsewhere. Modules for constructing a spatial information database and finding the possible faces on a contour are substantially complete; modules for the analysis of convexity, symmetry and cleavage are partially complete; false corner analysis has not been addressed, but its implementation is expected to be straightforward. A possible control strategy for the overall analysis system was outlined: this has not yet been implemented.

It should be noted that a primary aim of this study was the investigation of the problems involved in expanding the scope of petrographic image analysis and an exploration of the feasibility of extending image understanding to this domain, and not simply the creation of new software packages. In this respect the study has been successful.

The analysis of polycrystalline rock sections was expected at first to present a relatively simple image understanding task. One finding of this study, however, has been that petrographic images incorporate many of the fundamental problems to be found in image understanding, and certainly represent a respectable challenge for such investigations.

Practical experience within this problem domain has demonstrated some of the difficulties of implementing what are essentially perceptual tasks. For example, the contrast between the (at first sight) similar problems of finding a polygonal approximation for a contour and finding the significant linear structure on that contour turned out to be considerable. Further, to be useful in practical situations, using real data, analyses need to cope with departures from ideal form - in symmetry, convexity or collinearity - and the problem of finding the appropriate scale. In these respects, established image processing techniques were often found to be, basically, of little use.

It is considered that, building on the work presented above, a practically useful level of petrographic image understanding should be achievable, although with some difficulty. In the shorter term, however, it should certainly be possible to produce individual analysis 'modules' which would form useful additions to existing petrographic image analysis systems. In this regard, recent expressions of interest from the Petrophysics and Stone Conservation research group at the University of Oviedo have raised the possibility that their petrographic image analysis system may be a suitable host [Menendez 1991].

8.3 Further developments

It was noted above that parallels exist between petrographic image analysis and more general problems of computer vision; parallels between the principles involved in solving the assignment of boundaries problem and principles of perceptual grouping and figure-ground discrimination are particularly clear. Because such parallels exist, it was observed that the petrographic domain provides a suitable arena for the study of perceptual processes. The peculiarities of the domain impose constraints which make the problems more manageable, while they remain sufficiently challenging to support meaningful research. The existence of such parallels also means that solutions to problems of grouping and segmentation within this domain, and the assignment of boundaries problem in particular, can be expected to have more general applicability.

These ideas have been taken up by an interdisciplinary research group at the University of Birmingham: a formal proposal has been made to undertake a detailed study of the segmentation of petrographic images. The authors of the proposal are Dr E. Claridge (School of Computer Science), Professor G. W. Humphreys (School of Psychology) and Professor D. D. Hawkes (School of Earth Sciences). The proposal has been submitted to the ESRC / MRC / SERC Joint Council [Claridge *et al.* 1990]. Confidentiality does not permit a detailed description of the proposal to be given here; the following is a brief summary of the research aims:

> "... *Modelling of grouping mechanisms in the segmentation of visual scenes*
> This project will investigate how single perceptual grouping mechanisms of similarity, proximity, etc., work in combination to achieve scene segmentation of the kind conducted by human vision. Psychological experiments will provide initial hypotheses on how individual visual procedures interact. Based on these findings a computational model will be developed in which individual procedures will be implemented as rules and their precedence and mutual interactions as flexible control meta-rules. Further psychological experiments will involve the assessment of dynamic aspects of human performance, including the effects of different tasks and of different levels of expertise. The work will use images of petrographic, polycrystalline sections. Through the exploitation of the peculiarities of the petrographic domain, which impose particular physical constraints, perceptual processes will become more tractable. Solutions to the problems of grouping and segmentation within this limited domain will have general applicability. The developed successful procedures will be incorporated into a computer system capable of carrying out practical segmentation tasks."

This proposed research project can be viewed as an elaboration of the study reported in this thesis: the proposal does owe something to a draft of this thesis which was made available to the researchers, and a related publication [Thomson & Claridge 1989]. The aims of the two studies are similar, but the emphasis in the Birmingham proposal lies on psychological experimentation relating to the interaction of the component analyses.

It remains to be seen if the analysis rules presented in this thesis can be improved upon, or if the simple control structure offered here can be shown to be inadequate. The various analysis techniques developed in this study should, however, transfer easily to the proposed system. In general, the many findings, suggestions and useful practical techniques presented in this thesis can be expected to contribute substantially to this new research project.

REFERENCES

S.G.Akl, "Two remarks on a convex hull algorithm", *Information Processing Letters*, 8(2), pp108-109, 1979.

B.Allard & C.Sotin, "Determination of mineral phase percentages in granular rocks by image analysis on a microcomputer", *Computers & Geosciences*, **14**(2), pp261-269, 1988.

G.C.Amstutz, "A quantitative approach to paragenetic relations of ore minerals", *Freiberger Forschungshefte*, **C186** (Oelsner Festschrift), pp41-50, 1965.

G.C.Amstutz & H.Giger, "Metric and topological properties of rock and ore minerals", in Paulitsch, P. (Ed.) *Experimental and Natural Rock Deformation*, pp486-516, Berlin: Springer-Verlag, 1970.

G.C.Amstutz & H.Giger, "Stereological methods applied to mineralogy, petrology, mineral deposits and ceramics", *J. Microscopy*, **95**(1), pp145-164, Feb.1972.

D.H.Ballard, "Strip trees: a hierarchical representation for curves", *Commun. ACM*, **24**(5), pp310-321, 1981.

G.Barbery, "Mineral liberation analysis using stereological methods: a review of concepts and problems", in Park, W.C., Hausen, D.M., & Hagni, R.D. (Eds.), *Applied Mineralogy* (Proc. 2nd Int. Congress on Applied Mineralogy in the Minerals Industry, Los Angeles, 22-25 Feb. 1984), pp171-190, Warrendale, PA: The Metallurgical Soc. of AIME, 1984.

R.Barrett, A.Ramsay & A.Sloman, *POP-11: a practical language for Artificial Intelligence*, Chichester: Ellis Horwood, 1985.

H.G.Barrow & R.J.Popplestone, "Relational descriptions in picture processing", in Meltzer, B. & Michie, D. (Eds.), *Machine Intelligence* **6**, pp377-396, Edinburgh: Edinburgh University Press, 1971.

R.L.Bates & J.A.Jackson (Eds.), *Glossary of Geology* (2nd Edn.), Washington DC: Amer. Geol. Institute, 1980.

A.Bel-Lan & L.Montoto, "Stripe discrimination: application to boundary detection of twinned mineral grains in polarizing microscopy" (summary only), *Proc. 6th Int. Conf. on Pattern Recognition* (Oct. 1982, Munich), p1194, Silver Spring, MD: IEEE, 1982.

A.Bengtson & J-O.Eklundh, "Shape approximation by multiscale contour approximation", *Proc. 6th Scandinavian Conf. on Image Analysis: SCAI'89* (19-22 June, 1989, Oulu, Finland), pp252-261, 1989.

A.Bengtson, J-O.Eklundh & J.Howako, "Shape approximation by multiscale contour approximation", Technical Report TRITA-NA-8607, Dept. of Numerical Analysis & Computing Science, Royal Inst. of Technology, Stockholm, Sweden, 1986.

I.Biederman, H.J.Hilton & J.E.Hummel, "Pattern goodness and pattern recognition", in Pomerantz, J.R. & Lockhead, G.R. (Eds.), *The Perception of Structure*, Washington, DC: APA, 1991 (to appear).

R.C.Bolles, "Symmetry analysis of two-dimensional patterns for computer vision", *Proc. 6th Int. Jnt. Conf. on Artificial Intelligence: IJCAI-79* (Aug. 1979, Tokyo), pp70-72, Los Altos, CA: Wm. Kaufman, 1979.

R.C.Bolles & R.A.Cain, "Recognising and locating partially visible objects: the local-feature-focus method", in Pugh, A. (Ed.), *Robot Vision*, pp43-82, London: IFS, 1983.

J.A.Bondy & U.S.R.Murty, *Graph Theory with Applications*, London: Macmillan, 1976.

J.F.Boyce, G.A.Jones & V.F.Leavers, "An implementation of the Hough transform for line and circle location", *Proc. SPIE 4th Int. Symposium on Optics & Optoelectronic Science & Engineering* (30 March - 3 April, The Hague), 1987.

M.Brady, "Computational approaches to image understanding", *Computing Surveys*, **14**(1), pp3-71, March 1982.

M.Brady & H.Asada, "Smooth local symmetries and their implementation" in Brady, M. and Paul, R. (Eds.), *The First International Symposium on Robotics Research*, pp331-354, Cambridge, MA: MIT Press, 1984.

B.B.Chaudhuri & D.D.Majumder, "Recognition and fuzzy description of sides and symmetries of figures by computer", *Int. J. System Science*, **11**, pp1435-1445, 1980.

C.F.Chung, A.G.Fabbri, T.Kasvand & N.Otsu, "Grain contiguity and inference on permeability of granites", in Park, W.C., Hausen, D.M., & Hagni, R.D. (Eds.), *Applied Mineralogy* (Proc. 2nd Int. Congress on Applied Mineralogy in the Minerals Industry, Los Angeles, 22-25 Feb. 1984), pp223-244, Warrendale, PA: The Metallurgical Soc. of AIME, 1984.

E.Claridge, G.W.Humphreys & D.D.Hawkes, *Modelling of grouping mechanisms in segmentation of visual scenes*, (unpublished & confidential) research proposal submitted to ESRC / MRC / SERC Joint Council, Schools of Computer Science, Psychology and Earth Sciences, University of Birmingham, Dec. 1990.

M.W.Clark, "Quantitative shape analysis: a review", *Mathematical Geology* **13**(4), pp303-320, 1981.

N.N.Clark, "Three techniques for implementing digital fractal analysis of particle shape", *Powder Technology*, **46**, pp45-52, 1986.

E.F.Codd, "A relational model for large shared data banks", *Commun. ACM*, **13**, pp377-389, June 1970.

K.G.Cox, N.B.Price & B.Harte, *An Introduction to the Practical Study of Crystals, Minerals and Rocks* (revised 1st Edn.), London: McGraw-Hill, 1974.

S.J.Crabtree, R.L.Cannon & R.Ehrlich, "Image analysis in petrography", *Proc. IEEE Comp. Soc. Conf. on Computer Vision & Pattern Recognition*, pp504-507, Silver Spring, MD: IEEE, 1983.

S.J.Crabtree, R.Ehrlich & C.Prince, "Evaluation of strategies for segmentation of blue-dyed pores in thin sections of reservoir rocks", *Comput. Vision, Graphics & Image Process.*, **28**, pp1-18, 1984.

J.R.Craig, R-H.Yoon, R.M.Haralick, T-C.Pong & W-Z.Choi, "The application of the general image processing system (GIPSY) to mineral beneficiation studies", in Hagni, R.D. (Ed.), *Process Mineralogy II: applications in metallurgy, ceramics and geology* (Proc. Symposium held at AIME annual mtg, Dallas, TX, 14-18 Feb. 1982), pp55-67, Warrendale, PA: The Metallurgical Soc. of AIME, 1982.

G.W.Dagger, "Optical mineralogy using SIMPLIS", *Computers & Geosciences*, **16**(1), pp111-135, 1990.

L.S.Davis, "Understanding shape: angles and sides", *IEEE Trans. Comput.*, **C-26**(3), pp236-242, Mar. 1977a.

L.S.Davis, "Understanding shape II: symmetry", *IEEE Trans. Syst., Man, Cybern.*, **SMC-7**, pp204-212, Mar. 1977b.

R.Davis, B.G.Buchanan & E.H.Shortliffe, "Production rules as a representation in a knowledge-based consultation system", *Artificial Intelligence*, **18**, pp15-45, 1977.

R.Dearnley, "Effects of resolution on the measurement of grain 'size'", *Mineralogical Magazine*, **49**, pp539-546, Sept. 1985.

P.Delfiner, "A generalisation of the concept of size", *J.Microscopy*, **95**, pp203-216, 1972.

H.Dorans, *METEX: an expert system for metamorphic petrography*, PhD dissertation, Dept. of Geological Sciences, Aston University, Birmingham, 1988.

R.O.Duda & P.E.Hart, "Use of the Hough transform to detect lines and curves in pictures", *Commun. ACM*, **15**, pp11-15, Jan. 1972.

R.O.Duda & P.E.Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

J.G.Dunham, "Optimum uniform piecewise linear approximation of planar curves", *IEEE Trans. Patt. Anal. Mach. Intell.*, **PAMI-8**(1), pp67-75, Jan. 1987.

M.F.Dvali, D.S.Korzhinskii, Y.V.Linnik, M.A.Romanova & O.V.Sarmanov, "The life and work of A. B. Vistelius", in Romanova, M.A. & Sarmanov, O.V. (Eds.), *Topics in Mathematical Geology*, pp1-12, New York: Consultants Bureau Inc., 1970.

R.Ehrlich, S.K.Kennedy, S.J.Crabtree & R.L.Cannon, "Petrographic image analysis, I. Analysis of reservoir pore complexes", *J. Sedimentary Petrology*, **54**, pp1365-1378, Dec. 1984.

R.Ehrlich, T.Vogel, B.Weinberg, D.C.Kamilli, G.Byerly & H.Richter, "Textural variation in petrogenetic analysis", *Geol. Soc. of America Bulletin* **83**, pp665-676, Mar 1972.

R.Ehrlich & B.Weinberg, "An exact method for characterization of grain shape", *J. Sedimentary Petrology,* **40**, pp205-212, 1970.

A.G.Fabbri, *Image Processing of Geological Data*, New York: Van Nostrand Reinhold, 1984.

A.G.Fabbri, "Promising aspects of geological image analysis", in Hanley, J.T. & Merriam, D.F. (Eds.), *Microcomputer applications in geology*, pp199-213, Oxford: Pergamon, 1986.

A.G.Fabbri & T.Kasvand, "A picture processing approach to stereological problems", Mikroscopie, **37** (Supplement), pp431-436, 1980.

A.G.Fabbri, T.Kasvand & J.Masounave, "Adjacency relationships in aggregates of crystal profiles", in Haralick, R.M. (Ed.), *Pictorial Data Analysis* (Proc. NATO Advanced Study Inst., Bonas, France, 1-12 Aug. 1982), pp449-468, Berlin-Heidelberg: Springer-Verlag, 1983.

M.A.Fischler & R.C.Bolles, "Perceptual organization and curve partitioning", *IEEE Trans. Pattern Anal. Mach. Intell.*, **PAMI-8**(1), Jan. 1986.

D.Flinn, "Grain contacts in crystalline rocks", *Lithos*, **2**, pp361-370, 1969.

H.Freeman, "Boundary encoding and processing" in Lipkin, B.S. & Rosenfeld, A. (Eds.), *Picture Processing and Psychopictorics*, pp381-393, New York: Academic, 1970.

H.Freeman, "Computer processing of line-drawing images", *Computing Surveys*, **6**(1), pp57-97, March 1974.

H.Freeman, "Shape description via the use of critical points", *Pattern Recognition*, **10**, pp159-166, 1978.

R.M.Haralick, "A measure for circularity of digital figures", *IEEE Trans. Syst., Man, Cybern.*, **SMC-4**(4), pp394-396, July 1974.

R.M.Haralick, S.Krusemark & K.Neikirk, "GIPSY: introduction", Report SDA 81-4, Spatial Data Analysis Lab., Virginia Polytechnical Institute and State University, 1981.

R.M.Haralick & K.Shanmugan, "Computer classification of reservoir sandstones", *IEEE Trans. Geosci. Electron.*, **GE-11**, pp171-177, Oct. 1973.

R.M.Haralick, K.Shanmugan & I.Dinstein, "Textural features for image classification", *IEEE Trans. Syst., Man, Cybern.*, **SMC-6**, pp610-621, Nov. 1973.

R.M.Haralick, S.R.Sternberg & X.Zhuang, "Image analysis using mathematical morphology", *IEEE Trans. Patt. Anal. Mach. Intell.*, **PAMI-9**(4), pp532-550, July 1987.

R.M.Haralick & L.Watson, "A facet model for image data", *Comp. Graphics & Image Process.*, **15**, pp113-129, 1981.

F.H.Hatch, A.K.Wells & M.K.Wells, *Petrology of the Igneous Rocks*, London: Allen & Unwin, 1972.

P.G.Hoel, S.C.Port & C.J.Stone, *Introduction to Stochastic Processes*, Boston, MA: Houghton-Mifflin, 1972.

B.K.P.Horn, "Sequins and quills - representations for surface topography" in Bajcsy, R. (Ed.), *Representation of three-dimensional objects*, Berlin: Springer-Verlag, 1982.

E.Horowitz & S.Sahni, *Fundamentals of Data Structures in Pascal*, London: Pitman, 1984.

J.Illingworth & J.Kittler, "A survey of the Hough transform", *Comput. Vision, Graphics & Image Process.*, **44**, pp87-116, 1988.

R.Jarvis, "On the identification of the convex hull of a finite set of points in the plane", *Information Processing Letters*, **2**(1), pp18-21, 1973.

E.Johnston & A.Rosenfeld, "Angle detection on digital curves", *IEEE Trans. Comput.*, C-**24**, pp1006-1010, 1973.

M.P.Jones, "Automatic image analysis", in Zussman, J. (Ed.), *Physical methods in determinative mineralogy*, pp167-199, London: Academic, 1977.

M.P.Jones, "Application of electron probe microanalysis in the mineral industry", *Sampling and Analysis for the Minerals Industry* (Proc. Symposium organised by Inst. of Mining & Metallurgy, London, 2nd Nov. 1982), pp27-37, London: The Institution of Mining and Metallurgy, 1982.

M.P.Jones, "Recent developments in the rapid collection of quantitative mineralogical data ", in Park, W.C., Hausen, D.M., & Hagni, R.D. (Eds.), *Applied Mineralogy* (Proc. 2nd Int. Congress on Applied Mineralogy in the Minerals Industry, Los Angeles, 22-25 Feb. 1984), pp141-156, Warrendale, PA: The Metallurgical Soc. of AIME, 1984.

M.P.Jones, *Applied Mineralogy: a quantitative approach*, London: Graham and Trotman, 1987.

T.Kasvand, A.G.Fabbri & L.D.Nel, "Digitization and processing of large regional geological maps", Nat. Res. Council of Canada, Elect. Eng. Div., Report ERB-938, 1981.

W.Köditz, "A new powerful system for interactive and automatic image analysis", *Practical Metallography*, **18**(3), pp105-112, 1981.

R.Kretz, "Interpretation of the shape of mineral grains in metamorphic rocks", *J. Petrology*, **7**(1), pp68-94, 1966.

R.Kretz, "On the spatial distribution of crystals in rocks", *Lithos*, **2**, pp39-66, 1969.

Y.Kurozumi & W.A.Davis, "Polygonal approximation by the minimax method", *Comput. Graphics & Image Process.*, **19**, pp248-264, 1982.

J.P.Latham & A.B.Poole, "The application of shape descriptor analysis to the study of aggregate wear", *Quarterly Journal of Engineering Geology*, **20**, pp297-310, 1987.

J-J.Leou & W-H.Tsai, "Automatic rotational symmetry determination for shape analysis", *Pattern Recognition*, **20**(6), pp571-582, 1987.

M.D.Levine, "Region analysis using a pyramid data structure", in Tanimoto, S. & Klinger, A. (Eds.), *Structured Computer Vision*, pp57-100, New York: Academic, 1980.

M.D.Levine, *Vision in Man and Machine*, New York: McGraw-Hill, 1985.

C.Lin & M.H.Cohen, "Quantitative methods for microgeometric modelling", *J.Appl.Phys.*, **53**, pp4152-4165, June 1982.

D.G.Lowe, *Perceptual Organisation and Visual Recognition*, Boston, MA: Kluwer Academic, 1985

D.G.Lowe, "Three-dimensional object recognition from single two-dimensional images", *Artificial Intelligence*, **31**, pp355-395, 1987.

D.G.Lowe, "Organisation of smooth image curves at multiple scales", *Proc. 2nd Int. Conf. on Computer Vision* (Dec. 1988, Tampa, FL), pp558-567, Washington DC: IEEE Comput. Soc. Press, 1988.

D.G.Lowe & T.O.Binford, "Perceptual organisation as a basis for visual recognition", *Proc. 3rd Nat. Conf. on Artificial Intelligence: AAAI-83* (Aug. 1983, Washington DC), pp255-260, Los Altos, CA: Wm. Kaufman, 1983.

W.S.MacKenzie, C.H.Donaldson & C.Guilford, *Atlas of igneous rocks and their textures*, London: Longman, 1982.

W.S.MacKenzie & C.Guilford, *Atlas of rock-forming minerals in thin section*, London: Longman, 1980.

B.B.Mandelbrot, *The Fractal Geometry of Nature*, San Francisco: Freeman, 1982.

J.Martin, *Computer Data Base Organisation*, Englewood Cliffs, NJ: Prentice-Hall, 1977.

G.Matheron, *Elements pour une theorie des milieux poreux*, Paris: Masson et Cie., 1967

B.Menendez, personal communication, Jan. 1991.

D.L.Milgram, "Constructing trees for region description", *Comput. Graphics & Image Process.*, **11**, pp88-99, 1979.

F.Mokhtarian & A.K.Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes", *IEEE Trans. Pattern Anal. Machine Intell.*, **PAMI-8**(1), pp34-43, Jan.1986.

L.Montoto, "Image analysis methods on quantification of linear features", *Proc. IBM Int. Conf. on Image Processing & Pattern Recognition* (IBM Research Center, San Jose, CA), pp43.1-7, 1978.

L.Montoto, "Digital multi-image analysis: application to the quantification of rock microfractography", *IBM J. Res. Develop.*, **26**, pp735-745, Nov. 1982.

M.Montoto, *Principales actividades del grupo de investigacion "Petrofisica y Alteracion" (1980-1989)* (in Spanish), Area de Petrologia y Geoquimica, Departamento de Geologia, Universidad de Oviedo, Spain, Aug. 1990.

M.Montoto, A.Bel-Lan & L.Montoto, "Microscopic quantification of textures and fissures in rocks by digital image processing", *Proc. 3rd Int. Cong. Int. Assoc. Eng. Geol.*, Sec. II, Vol.2, pp51-60, Madrid: Servicio Geologico de O.P., 1978.

G.A.Moore, "Automatic scanning and computer processes for the quantitative analysis of micrographs and equivalent subjects", in Cheng, G.C., Ledley, R.S., Pollock, D.K. & Rosenfeld, A. (Eds.), *Pictorial pattern recognition* (Proc. Symposium on Automatic Photointerpretation, 31 May - 2 June 1967, Washington DC), pp275-326, Washington DC: Thompson, 1968.

R.Nawrath & J.Serra, "Quantitative image analysis: theory and instrumentation", *Microscopica Acta*, **82**(2), pp101-111, Sept. 1979a.

R.Nawrath & J.Serra, "Quantitative image analysis: applications using sequential transformations", *Microscopica Acta*, **82**(2), pp113-128, Sept. 1979b.

A.M.Nazif, *A rule-based expert system for image segmentation*, PhD dissertation, Dept. of Elec. Eng., McGill University, Montreal, Canada, 1983.

C.J.Oddy, A.J.Rye & R.D.Tavendale, "Software system design for general purpose image analysis", *GEC Journal of Research*, **1**(1), pp48-58, 1983.

H.Oettel & J.Ohser, "AMBA/R-TECH - software for the structural and textural analysis of materials", *Neue Technische Büro*, 3(3), pp92-96, 1986.

Y.Ohta, *Knowledge-based interpretation of outdoor natural colour scenes*, Boston, MA: Pitman, 1985.

T.E.Orford & W.B.Whalley, "The use of the fractal dimension to quantify the morphology of irregular-shaped particles", *Sedimentology*, **30**, pp655-668, 1983.

T.Pavlidis, *Structural Pattern Recognition*, Berlin-Heidelberg-New York: Springer-Verlag, 1977.

T.Pavlidis, "Curve fitting as a pattern recognition problem, *Proc 6th Int. Conf. on Pattern Recognition* (Oct. 1982, Munich), pp853-858, Silver Spring, MD: IEEE, 1982.

T.Pavlidis & S.L.Horowitz, "Segmentation of plane curves", *IEEE Trans. Comput.*, **C-23**, pp860-870, Aug. 1974.

W.A.Perkins, "Using circular symmetry and intensity profiles for computer vision inspection", *Comput. Graphics & Image Process.*, **17**, pp161-172, 1981.

D.Petkovic "The need for accuracy verification of machine vision algorithms and systems", *Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition* (4-8 June 1989, San Diego, CA), pp439-440, Washington DC: IEEE Comp. Soc. Press, 1989.

W.Petruk, "Image analysis in process mineralogy", in Hagni, R.D. (Ed.), *Process Mineralogy II: applications in metallurgy, ceramics and geology* (Proc. Symposium held at AIME annual mtg, 14-18 Feb. 1982, Dallas, TX), pp39-53, Warrendale, PA: The Metallurgical Soc. of AIME, 1982.

W.Petruk, "The capabilities of the microprobe Kontron image analysis system: application to mineral beneficiation", *Scanning Microscopy*, **2**(3), pp1247-1256, 1988.

F.C.Phillips, *An Introduction to Crystallography* (4th Edn.), London: Longmans, 1971.

T-Y.Phillips, L.S.Davis & A.Rosenfeld, "Automatic segmentation of electron micrographs of Berea sandstone cross-sections", *Pattern Recognition*, **16**, pp385-400, 1983.

T-Y.Phillips & A.Rosenfeld, "A method of curve partitioning using arc-chord distance", *Pattern Recognition Letters*, **5**(4), pp285-288, Apr. 1987.

T-C.Pong, R.M.Haralick, J.R.Craig, R-H.Yoon & W-Z.Choi, "The application of image analysis techniques to mineral processing", *Pattern Recognition Letters*, **2**, pp117-123, Dec. 1983.

U.Ramer, "An iterative procedure for the polygonal approximation of plane curves", *Comput. Graphics & Image Process.*, **1**, pp244-256, 1972.

A.Rosenfeld, "Computer vision: basic principles", *Proc. IEEE*, **76**(8), pp863-868, 1988.

A.Rosenfeld & A.C.Kak, *Digital Picture Processing* (2nd Edn.), New York: Academic, 1982.

A.Rosenfeld & R.A.Melter, "Digital geometry", *Mathematical Intelligencer*, **11**(3), pp69-72, 1989.

A.Rosenfeld & J.L.Pfaltz, "Sequential operations in digital picture processing", *J. ACM*, **13**(4), pp471-494, Oct.1966.

P.L.Rosin & G.A.W.West, "Detection of circular arcs in images", *Proc. 4th Alvey Vision Conf.: AVC88* (31 Aug. - 2 Sept., Manchester), pp259-263, 1988.

H.Samet, "Region representation: quadtrees from binary arrays", *Comput. Graphics & Image Process.*, **27**, pp88-93, 1980.

R.Sedgewick, *Algorithms*, Reading, MA: Addison-Wesley, 1983.

J.Serra, *Image Analysis and Mathematical Morphology*, London: Academic, 1982.

J.Sklansky, "Recognition of convex blobs", *Pattern Recognition*, **2**, pp3-10, 1970.

S.Suzuki & K.Abe, "Topological structural analysis of digitized binary images by border following", *Comput. Vision, Graphics, & Image Process.*, **30**, pp32-46, 1985.

S.L.Tanimoto, *Shape measures for polygons based upon the distributions of angles, slopes and lengths of sides*, Technical Report 80-01-01, Dept. of Computer Science, University of Washington, Seattle, 1980.

R.C.Thomson, *Convexity in petrographic image analysis,* Technical Report TR91009, Dept. of Computer Science & Applied Mathematics, Aston University, Birmingham, 1991.

R.C.Thomson & E.Claridge, "A 'computer vision' approach to the analysis of crystal profiles in thin sections", *Proc. 6th Scandinavian Conf. on Image Analysis* (19-22 June 1989, Oulu, Finland), pp1208-1215, 1989.

R.C.Thomson & E.Sokolowska, "Mineral cleavage analysis via the Hough transform", in Kittler, J. (Ed.), *Pattern Recognition 1988* (Proc. BPRA 4th Int. Conf. on Pattern Recognition, 28-30 March 1988, Cambridge), pp390-398, London: Springer-Verlag, 1988.

I.Tomek, "Two algorithms for piecewise linear continuous approximations of functions of one variable", *IEEE Trans. Comput.*, **C-23**, pp445-448, Apr. 1974.

E.E.Underwood, *Quantitative Stereology*, Reading, MA: Addison-Wesley, 1970.

A.B.Vistelius, *Studies in Mathematical Geology*, New York: Consultants Bureau Inc., 1967.

G.S.Watson, "Texture analysis", *Geol. Soc. of America Memoir* **142**, pp367-391, 1975.

H.Wechsler, "A structural approach to shape analysis using mirroring axes", *Comput. Graphics & Image Process.*, **9**, pp246-266, 1979.

N.Weisstein & E.Wong, "Figure-ground organisation and the spatial and temporal responses of the visual system", in Schwab, E.C. & Nusbaum, H.C. (Eds.), *Pattern Recognition by Humans and Machines, Volume 2: Visual Perception*, pp31-64, London: Academic, 1986.

J.West, "Towards an expert system for identification of minerals in thin section", *J. International Assoc. for Mathematical Geology*, **17**(7), pp743-753, Oct. 1985.

E.H.T.Whitten & M.F.Dacey, "On the significance of certain Markovian features of granite textures", *J. Petrology*, **16**, pp429-453, 1975.

C.M.Williams, "An efficient algorithm for the piecewise linear approximation of planar curves", *Comput. Graphics & Image Process.*, **8**, pp286-293, 1978.

H.Williams, F.J.Turner & C.M.Gilbert, *Petrography. An introduction to the study of rocks in thin sections* (2nd Edn.), San Francisco: Freeman, 1982.

A.P.Witkin, "Scale space filtering", *Proc. 8th Int. Jnt. Conf. on Artificial Intelligence: IJCAI-83* (Aug. 1983, Karlsruhe), pp1019-1022, Los Altos, CA: Wm. Kaufman, 1983.

L.A.Zadeh, "Fuzzy sets", *Information & Control* **8**, pp338-353, 1965.